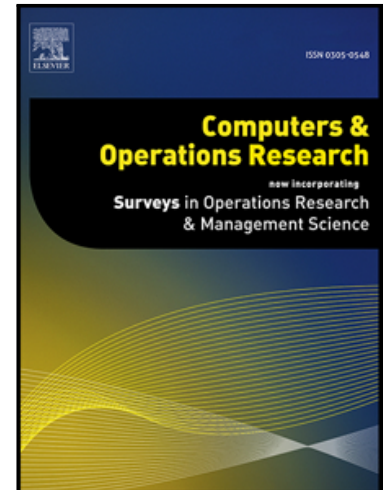


Accepted Manuscript

A two-stage approach to the UCITS-constrained index-tracking problem

O. Strub , N. Trautmann

PII: S0305-0548(18)30254-5
DOI: <https://doi.org/10.1016/j.cor.2018.10.002>
Reference: CAOR 4562



To appear in: *Computers and Operations Research*

Received date: 21 February 2018
Revised date: 15 August 2018
Accepted date: 1 October 2018

Please cite this article as: O. Strub , N. Trautmann , A two-stage approach to the UCITS-constrained index-tracking problem, *Computers and Operations Research* (2018), doi: <https://doi.org/10.1016/j.cor.2018.10.002>

This is a PDF file of an unedited manuscript that has been accepted for publication. As a service to our customers we are providing this early version of the manuscript. The manuscript will undergo copyediting, typesetting, and review of the resulting proof before it is published in its final form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

Highlights

- We study the index-tracking problem subject to UCITS regulations.
- We propose a new mixed-integer quadratic programming formulation of this problem.
- We develop a heuristic based on a genetic algorithm and local branching.
- We present a new representation of individuals for the genetic algorithm.
- We find that the UCITS regulations reduce the out-of-sample portfolio risk.

A two-stage approach to the UCITS-constrained index-tracking problem

O. Strub^{1,*}, N. Trautmann¹

¹*Department of Business Administration, University of Bern, Schützenmattstrasse 14, 3012 Bern, Switzerland*

Abstract

Undertakings for Collective Investments in Transferable Securities (UCITS) are investment funds that are regulated by the European Union. UCITS have become increasingly popular, resulting in a total corresponding amount of assets under management of €8.5 trillion by the end of 2016. We present a two-stage approach to the problem of how to construct a portfolio of assets for a UCITS that aims to replicate the returns of a financial index subject to the constraints imposed by the UCITS regulations. In the first stage, we apply a genetic algorithm that treats subsets of the index constituents as individuals to construct a good feasible solution in a short CPU time. In this genetic algorithm, we use a new representation of subsets, which is the first to exhibit all of the following four desirable properties: feasibility, efficiency, locality, and heritability. In the second stage, we apply local branching based on a new mixed-integer quadratic programming formulation to improve the best solution obtained in the first stage. In a numerical experiment on real-world data, the approach yields very good feasible solutions in a short CPU time.

Keywords: Portfolio management, Index tracking, Mixed-integer quadratic programming, Heuristics

1. Introduction

An investment fund is a pool of capital collected from different investors. Professional asset managers invest the collected capital on behalf of the investors in a portfolio of assets such as stocks or bonds. Investment funds that aim to replicate or track the returns of a particular financial index are known as index funds. Index funds are very popular because, compared with investment funds that aim to achieve an excess return over an index, they are less expensive to manage, which translates into lower fees for the investors, and they often yield higher returns (cf., e.g., Busse et al. [10], Malkiel [33], Montfort et al. [35]). To achieve a small tracking error when replicating index returns, the most intuitive approach is full replication, which requires an investment in all constituents of an index in accordance with the index composition. One drawback of full replication is the high management and transaction costs that arise for indices with many constituents (cf., e.g., Guastaroba and Speranza [25], Sharma et al. [50]). By investing in only a small subset of the index constituents, these costs can be reduced substantially.

*Corresponding author

Email addresses: `oliver.strub@pqm.unibe.ch` (O. Strub), `norbert.trautmann@pqm.unibe.ch` (N. Trautmann)

Undertakings for Collective Investments in Transferable Securities (UCITS) are investment funds that are regulated by the European Union (EU). UCITS have become economically important in recent years, and over €8.5 trillion in net assets were managed through such funds at the end of 2016 (cf. European Fund and Asset Management Association (EFAMA) [17]); this is comparable to the US \$16 trillion scale of the US mutual fund industry (cf. Investment Company Institute [26]). UCITS are subject to regulatory constraints imposed by the UCITS directive of the European Parliament. As noted by Kolm et al. [27], such regulatory constraints may present a challenge for asset managers when constructing their portfolios.

We consider the UCITS-constrained index-tracking problem (UCITP) introduced by Krink et al. [29], which is the problem of how to construct a new portfolio from cash for a UCITS index fund, i.e., an index fund regulated by the EU. The objective of the UCITP is to minimize the mean-squared error (MSE) between the returns of the portfolio and the index over a set of historical in-sample periods. The MSE is one of the most widely used measures of the tracking error in practice (cf. Corielli and Marcellino [15]) and in the literature (cf., e.g., Andriosopoulos and Nomikos [2], Beasley et al. [4], Chiam et al. [14], Maringer and Oyewumi [34], Montfort et al. [35], Sant'Anna et al. [47]). The underlying assumption motivating the minimization of the MSE for historical in-sample periods is that a small in-sample MSE will also tend to lead to a small MSE in out-of-sample periods. The UCITP comprises the following constraints. A lower and an upper bound on the number of different assets that can be included in the portfolio are prescribed. In addition, a lower bound on the relative weight of each asset selected for inclusion in the portfolio is prescribed. Finally, the constraints of the UCITS directive must be satisfied. These include a short-selling prohibition and the 5/10/40 concentration rule, which states that the weight of each selected asset must not exceed a lower threshold of 5%, except that the weights of some assets may be increased up to a middle threshold of 10%, provided that the sum of the weights exceeding the lower threshold does not exceed an upper threshold of 40%.

In the literature, two approaches to the UCITP have been proposed: a mixed-integer quadratic programming (MIQP) approach (cf. Scozzari et al. [49]) and an approach based on differential evolution and combinatorial search (cf. Krink et al. [29]). Both approaches yield good feasible solutions to small- and medium-scale instances of the UCITP, but fail to do so for large-scale instances. The reason for this is the substantial amount of CPU time required for fine-tuning the portfolio weights by applying differential evolution and for solving the quadratic-programming relaxations. For other optimization problems in finance, genetic algorithms have previously been successfully applied (cf. Gilli and Schumann [22]). Specifically, several genetic algorithms have been proposed for solving the index-tracking problem without the UCITS regulatory constraints (cf., e.g., Andriosopoulos and Nomikos [2], Beasley et al. [4], Chiam et al. [14], Ruiz-Torrubiano and Suárez [45]). According to Gottlieb et al. [24], the most important element in the design of such genetic algorithms is the representation, i.e., the mapping between the data structure of a solution, referred to as the genotype, and the decoded solution, referred to as the phenotype. To enable the design of an efficient and effective genetic algorithm, the representation should exhibit the following four properties

(cf. Gottlieb et al. [24]): efficiency, meaning that a genotype can be rapidly decoded into its corresponding phenotype; locality, meaning that small changes in a genotype lead to small changes in the corresponding phenotype; heritability, meaning that combining parent genotypes using crossover operators produces child genotypes whose corresponding phenotypes exhibit combined features of the parent phenotypes; and a fourth property that is called feasibility hereafter. The feasibility property is satisfied if all feasible and no infeasible phenotypes are represented in the set of all possible genotypes. To the best of our knowledge, there is no representation of subsets in the literature that exhibits the feasibility property with respect to a constraint on the subset's minimum and maximum cardinality. Moreover, the existing genetic algorithms lack the ability to handle the regulatory constraints for UCITS.

The contributions of this paper are threefold. First, we present a new representation of subsets that exhibits the four desired properties of efficiency, locality, heritability, and feasibility. The feasibility property enables the use of fast and simple conventional evolutionary operators without requiring any time-consuming repair operators or penalty functions to handle infeasible phenotypes. The proposed representation of subsets should be of general interest because it can be used in genetic algorithms for any optimization problem that involves the selection of a subset, such as the feature-selection problem in machine learning. Second, we present a new MIQP formulation of the UCITP that requires fewer constraints than the existing formulation of Scozzari et al. [49]. Third, we present a new two-stage approach to the UCITP that is able to devise very good feasible solutions for UCITP instances of arbitrary size in a short CPU time. In the first stage of the proposed approach, we simplify the UCITP by considering only equally weighted portfolios, which results in a pure combinatorial asset-selection problem. To solve this asset-selection problem, we apply a genetic algorithm based on the proposed subset representation. The purpose of the first stage is to obtain a good feasible solution in a short CPU time. In the second stage, we improve the best solution obtained in the first stage by applying a local-search method based on mixed-integer programming using the new MIQP formulation of the UCITP and the local-branching framework that was introduced by Fischetti and Lodi [19]. Compared with other local-search methods, the advantage of local branching is that it is exact in nature, which allows provably optimal solutions to be determined starting from any feasible initial solution for small problem instances, but can also be applied heuristically, which allows very good feasible solutions to be determined for larger instances.

We report a computational experiment performed using 45 UCITP instances based on real-world stock-market data. The four main results of this experiment are as follows: 1) in comparison with two existing subset representations, using the new representation instead in the genetic algorithm leads to faster evolution and better results in terms of the objective function value; 2) when an MIQP approach is applied subject to a limit on the CPU time, the new MIQP formulation leads to better results in terms of the objective function value than the existing MIQP formulation; 3) the two-stage approach leads to better results than the MIQP approach based on the new MIQP formulation and a genetic algorithm based on the proposed subset representation within a limited CPU time in terms of the objective function value and of the out-of-

sample tracking error; and 4) the UCITS regulations reduce the portfolio risk in terms of the out-of-sample tracking error.

The remainder of this paper is organized as follows. In Section 2, we review the related literature. In Section 3, we present the proposed two-stage approach. In Section 4, we report the results of our computational experiment. In Section 5, we offer some concluding remarks and an outlook on future research.

2. Related literature

In this section, we review the related literature. In Subsection 2.1, we present an overview of existing approaches to the index-tracking problem without regulatory constraints. In Subsection 2.2, we present the MIQP formulation of the UCITP introduced by Scozzari et al. [49]. In Subsection 2.3, we discuss the existing subset representations used in genetic algorithms.

2.1. Index tracking

Sharma et al. [50] categorize index-tracking approaches into two broad groups. Approaches in the first group use factor models to construct a portfolio (cf., e.g., Canakgoz and Beasley [11], Corielli and Marcellino [15], Rudd [42]). Approaches in the second group minimize some measure of the tracking error, often subject to a cardinality constraint, i.e., a constraint on the number of assets that can be included in the portfolio. Here, we focus on the second group of approaches, specifically on the different tracking-error measures that have been used. These measures can themselves be divided into two groups: value-based and return-based tracking-error measures (cf. Gaivoronski et al. [20], Strub and Baumann [54]).

Return-based tracking errors are calculated based on the returns of the portfolio and the index. Roll [40] minimizes the tracking-error variance (TEV), i.e., the variance of the differences between the portfolio returns and the index returns. Kwiatkowski [31] minimizes the TEV subject to a cardinality constraint. Mutunge and Haugland [37] show that the latter problem is NP-hard, and present a greedy heuristic to tackle the problem. The TEV is commonly used in both practical and theoretical work (cf. Corielli and Marcellino [15]). Nevertheless, Beasley et al. [4] argue against the use of the TEV because with this measure, a portfolio can have a tracking error of zero even if its returns are constantly below those of the index. Because of this drawback of the TEV, the mean-squared error (MSE) of the return differences has often been used instead in the literature (cf., e.g., Andriosopoulos and Nomikos [2], Beasley et al. [4], Benidis et al. [5], Chiam et al. [14], Maringer and Oyewumi [34], Sant'Anna et al. [48], Takeda et al. [57]). According to Rudolf et al. [43], the use of quadratic tracking-error measures such as the TEV and the MSE is common in financial practice because they exhibit a number of desirable statistical properties. However, Rudolf et al. [43] argue that quadratic tracking-error measures are difficult for practitioners to interpret, and they propose four different tracking-error measures based on the absolute differences between the returns of the portfolio and the index. One advantage of these four measures is that they can be formulated as linear objective functions. Chen

and Kwon [12] maximize the correlation between the portfolio returns and the index returns, which is also formulated as a linear objective function.

Value-based tracking errors are calculated based on the value developments of the portfolio and the index. Konno and Wijayanayake [28] and Guastaroba and Speranza [25] use the mean absolute deviation (MAD) between the value developments of the portfolio and the index as a measure of the tracking error. The MAD can also be formulated as a linear objective function. Strub and Baumann [54] propose a value-based tracking error that also exhibits properties of return-based tracking errors; specifically, the proposed tracking error is zero if and only if the historical returns of the portfolio and the index coincide.

Following Krink et al. [29], who introduced the UCITP, we chose to use the MSE in this paper because it is very commonly applied in practice and in the literature. However, the approach presented in this paper could be used without structural adjustments for any tracking-error measure that can be formulated as a linear or a convex quadratic function.

2.2. Existing MIQP formulation of the UCITP

The MIQP formulation (M-STPK) of the UCITP presented by Scozzari et al. [49] is given below. Table 1 defines the nomenclature used in this MIQP formulation.

$$\begin{aligned}
 & \left\{ \begin{array}{ll} \text{Min.} & \frac{1}{|T|} \sum_{t \in T} \left(\sum_{i \in I} r_t^i w_i - r_t \right)^2 & (1) \\ \text{s.t.} & \sum_{i \in I} w_i = 1 & (2) \\ & l \leq \sum_{i \in I} y_i \leq k & (3) \\ & \varepsilon y_i \leq w_i \leq \delta y_i & (i \in I) \quad (4) \\ & \sum_{i \in I} v_i \leq \eta & (5) \\ & \zeta u_i \leq w_i \leq \zeta + u_i & (i \in I) \quad (6) \\ & w_i + u_i - 1 \leq v_i \leq w_i & (i \in I) \quad (7) \\ & v_i \leq u_i & (i \in I) \quad (8) \\ & w_i \geq 0, v_i \geq 0, y_i \in \{0, 1\}, u_i \in \{0, 1\} & (i \in I) \quad (9) \end{array} \right. & \text{(M-STPK)}
 \end{aligned}$$

The objective function given in (1) captures the MSE between the returns of the portfolio ($\sum_{i \in I} r_t^i w_i$) and the corresponding index (r_t) over all historical periods $t \in T$. Constraint (2) is the budget constraint and ensures that the portfolio weights sum up to one. The cardinality constraint (3) defines a feasible range between l and k for the number of assets to be included in the portfolio. The cardinality constraint employs binary variables y_i , where y_i is equal to one if asset $i \in I$ is included in the portfolio and zero otherwise. The constraints defined in (4) impose a lower bound (ε) and an upper bound that corresponds to the middle UCITS threshold (δ) on the weight of each asset included in the portfolio and simultaneously ensure that the

Table 1: Nomenclature for the MIQP formulation of Scozzari et al. [49].

<i>Parameters and sets:</i>	
n	Number of index constituents
I	Set of identity tags of the index constituents ($I = \{1, \dots, n\}$)
T	Set of historical in-sample time periods
l/k	Minimum/maximum portfolio cardinality
ε	Minimum weight of each asset if selected
$\zeta/\delta/\eta$	Lower/middle/upper UCITS concentration-rule thresholds
r_t/r_t^i	Return of index/asset $i \in I$ in period $t \in T$
<i>Decision variables:</i>	
w_i	Weight of asset $i \in I$ in the portfolio
y_i	$= 1$, if $w_i > 0$; $= 0$, otherwise ($i \in I$)
v_i	$= w_i$, if $w_i > \zeta$; $= 0$, otherwise ($i \in I$)
u_i	$= 1$, if $w_i > \zeta$; $= 0$, otherwise ($i \in I$)

binary variables y_i are assigned the appropriate values. Constraint (5) limits the sum of the portfolio weights that exceed the lower UCITS threshold (ζ) to the upper UCITS threshold (η). In this constraint, continuous decision variables v_i are used, where v_i is equal to the weight of asset i if its weight exceeds the lower UCITS threshold and zero otherwise. To determine appropriate values of these continuous decision variables v_i , binary decision variables u_i are introduced, where u_i is equal to one if the weight of asset i exceeds the UCITS lower threshold and zero otherwise. Appropriate values are assigned to these binary decision variables based on the constraints defined in (6). Based on the values of the binary decision variables u_i , the constraints defined in (7) assign appropriate values to the continuous decision variables v_i . The constraints given in (8) ensure that each variable v_i is set to zero if the weight of asset i does not exceed the lower UCITS threshold, i.e., $u_i = 0$. The domains of the decision variables are specified by (9).

2.3. Existing subset representations

In this subsection, we discuss the existing representations of subsets that have previously been used in genetic algorithms for problems that involve the selection of a subset of a set of identity tags $I = \{1, \dots, n\}$ subject to a feasible range for the subset's cardinality. Examples of such problems are the UCITP considered in this paper, the index-tracking problem without regulatory constraints, and the problem of selecting the best features for linear regression or machine learning (cf., e.g., Bertolazzi et al. [6], Bertsimas and King [7], Bertsimas et al. [8]). The representations that have previously been used in genetic algorithms for solving these problems can be divided into two classes: pure subset representations and mixed representations. Phenotypes of the first class represent subsets only. By contrast, phenotypes of the second class also represent additional decisions related to the elements to be included in the subset, such as the portfolio weights.

These two classes of representations can each be further divided into two subclasses based on the genotypes used: pure subset representations comprise binary and integer representations, whereas mixed representations

comprise real-valued and hybrid representations. In the following, we describe the four subclasses.

In binary representations, a vector $\{0, 1\}^n$ is used as a genotype (cf. Brill et al. [9], Kuncheva and Jain [30], Moral-Escudero et al. [36], Oh et al. [38], Ruiz-Torrubiano and Suárez [44], Siedlecki and Sklansky [51]). The binary digits correspond to the decisions regarding whether each element is included in the subset. For example, if the i^{th} digit in the vector is equal to one, then the identity tag i is included in the subset.

In integer representations, the genotypes are based on integers that correspond to the identity tags of the selected elements. In Strub and Trautmann [55], a vector of distinct integers from the set I is used as a genotype. Moral-Escudero et al. [36] and Ruiz-Torrubiano and Suárez [45, 46] directly use subsets of the set I as genotypes.

In real-valued representations, a vector \mathbb{R}^n is employed as a genotype (cf. Andriosopoulos and Nomikos [2], Diosan [16], Streichert et al. [53]). A corresponding phenotype is constructed by including in the subset all identity tags i such that the value of the i^{th} element in the real-valued vector is non-zero. If identity tag i is included in the subset, then the i^{th} value in the real-valued vector is used as the value of the associated continuous decision variable.

Hybrid representations are combinations of either binary or integer representations with real-valued representations. Chiam et al. [14], Raymer et al. [39], Skolpadungket et al. [52], and Streichert et al. [53] use a binary vector $\{0, 1\}^n$ and a real-valued vector \mathbb{R}^n as a genotype. The value of the i^{th} element in the real-valued vector is multiplied by the value of the i^{th} element in the binary vector. If the resulting value is non-zero, then the identity tag i is included in the subset, and the resulting value is assigned to the associated continuous decision variable. Hence, the binary vector can be interpreted as a masking vector (cf. Raymer et al. [39]). Chiam et al. [13] consider the mean-variance portfolio-optimization problem and use a permutation of the vector $[1, 2, \dots, n]$ combined with a real-valued vector \mathbb{R}^n as a genotype. The portfolio is constructed by selecting the assets with the identity tags defined by the order of the permuted vector. The assets are included in the subset, with weights assigned in accordance with the values in the real-valued vector, until the sum of the weights of the assets included in the portfolio exceeds one. Then, all weights are normalized such that their sum is equal to one.

Table 2 presents an illustrative example of how the discussed representations are used to decode a genotype into the corresponding phenotype. For all representations except the integer representation, the table shows a possible genotype that is decoded into a subset with an infeasible cardinality. Moreover, the integer representation and the second hybrid representation listed in the table require special evolutionary operators that maintain the properties of the genotypes, i.e., the uniqueness of the integers in each genotype. Hence, none of the discussed representations exhibits the feasibility property, which means that either simple and fast conventional evolutionary operators cannot be used or penalty functions or repair operators must be applied to handle infeasible phenotypes.

Table 2: Illustrative example of the subset representations with $n = 5$, a feasible subset cardinality of three or four, and associated continuous variables that correspond to the portfolio weights w_i , $i = 1, \dots, n$.

Representation	Possible genotype	Decoded phenotype	Feasible
Binary	[0, 1, 0, 0, 1]	{2, 5}	✗
Integer	[2, 3, 5]	{2, 3, 5}	✓
Real	[0, 0.75, 0, 0, 0.25]	{2, 5}, $w_2 = 0.75$, $w_5 = 0.25$	✗
Hybrid with binary	[0, 1, 0, 0, 1], [0.5, 0.75, 0.8, 0.6, 0.25]	{2, 5}, $w_2 = 0.75$, $w_5 = 0.25$	✗
Hybrid with integer	[2, 5, 4, 3, 1], [0.35, 0.9, 0.8, 0.25, 0.3]	{2, 5}, $w_2 = 0.75$, $w_5 = 0.25$	✗

3. Solution approach

In Strub and Trautmann [56], we presented a preliminary version of the solution approach proposed in this paper. In this preliminary version, we used a hybrid genetic algorithm similar to that of Moral-Escudero et al. [36], in which the fitness of each individual is determined by applying an exact solution method such as mixed-integer programming. To reduce the CPU time for the fitness evaluations, we first estimated the fitness of the individuals in an efficient way and then evaluated the fitness of promising individuals only. In the present paper, we propose a new way of combining a genetic algorithm with mixed-integer programming in a sequential manner. Specifically, we present a two-stage approach in which a genetic algorithm is used in the first stage to determine a good feasible equally weighted portfolio and an MIQP-based local-branching method is used in the second stage to improve the solution from the first stage. The new two-stage approach produces superior results compared with the approach presented in Strub and Trautmann [56] and even allows provably optimal solutions to be determined for small-scale instances.

In Subsection 3.1, we present the new MIQP formulation that we use in the local-branching method. In Subsection 3.2, we present the new subset representation that we use in the genetic algorithm. In Subsection 3.3, we present the two-stage approach in detail. Table 3 defines the additional notation used.

3.1. New MIQP formulation of the UCITP

For the new MIQP formulation of the UCITP, we replace the continuous and binary decision variables v_i and u_i , respectively, that are used in Scozzari et al. [49] with the two kinds of decision variables x_i and z_i .

The non-negative continuous variables x_i are defined such that they must be assigned at least the difference between the weight of asset i and the lower UCITS threshold. The constraints defined in (10) determine the values of the variables x_i according to this definition.

$$w_i - \zeta \leq x_i \quad (i \in I) \quad (10)$$

The binary variables z_i are defined such that they must be equal to one if the weight of asset i exceeds the lower UCITS threshold and can be equal to either zero or one otherwise. The constraints defined in (11) determine the values of the binary variables z_i according to this definition. If the weight of asset i exceeds

Table 3: Additional nomenclature for the two-stage approach.

<i>Parameters and sets:</i>	
s	Size of population (number of individuals)
P	Population (set of individuals)
M	Mating pool (set of individuals)
\underline{d}/\bar{d}	Minimum/maximum dimension of genotype vectors, with $\underline{d} > 0$ and $\bar{d} \leq n$
$\mathbf{g}^i \in \{1, \dots, n\}^{d_i}$	Genotype vector $[g_1^i, \dots, g_{d_i}^i]$ of individual i with $d_i \in \{\underline{d}, \dots, \bar{d}\}$
$f(\mathbf{g}^i)$	Fitness of individual i
\mathbf{b}	Genotype vector of the individual with the best known fitness
<i>random</i>	Uniformly distributed random number from the half-closed interval $[0,1)$
p_c	Probability of crossover
$p_e/p_a/p_r$	Probability of an exchange/addition/removal of an element in/to/from a genotype vector during mutation
n_g	Maximum number of generations for the genetic algorithm
n_s	Maximum number of stocks that are considered during local branching ($k \leq n_s \leq n$)
<i>Decision variables:</i>	
x_i	$\geq w_i - \zeta$, if $w_i > \zeta$; ≥ 0 , otherwise ($i \in I$)
z_i	$= 1$, if $w_i > \zeta$; $\in \{0, 1\}$, otherwise ($i \in I$)

the lower UCITS threshold, then the non-negative continuous variables x_i are assigned a value that is greater than zero because of constraints (10). In this case, constraints (11) ensure that the binary variables z_i are assigned a value of one. By contrast, if the weight of asset i does not exceed the lower UCITS threshold, then the non-negative continuous variables x_i can be assigned a value of zero or greater than zero according to constraints (10). In this case, the binary variables z_i can take a value of either zero or one depending on the value of x_i according to constraints (11). In addition, the constraints given in (11) set upper bounds on the variables x_i because the weight of any asset cannot exceed the lower UCITS threshold by more than $\delta - \zeta$.

$$x_i \leq (\delta - \zeta)z_i \quad (i \in I) \quad (11)$$

Based on the non-negative continuous variables x_i and the binary variables z_i , the constraint given in (12) ensures that the sum of the weights exceeding the lower UCITS threshold does not exceed the upper UCITS threshold. The left-hand side of constraint (12) corresponds only to an upper bound on the sum of the weights of the assets whose weights exceed the lower UCITS threshold because the non-negative continuous variables x_i correspond to merely an upper bound on the difference between the weight of asset i and the lower UCITS threshold and because the binary variables z_i can have a value of one even if $w_i \leq \zeta$. However, to ensure that the 5/10/40 UCITS concentration rule is satisfied, it is sufficient to use an upper bound on the sum of the weights of the assets whose weights exceed the lower UCITS threshold. To see this, consider a portfolio that satisfies the 5/10/40 UCITS concentration rule. Then, the variables x_i and z_i can be assigned appropriate values such that the constraints (10)–(12) are satisfied. By contrast, if a portfolio does not satisfy the 5/10/40 UCITS concentration rule, then it is impossible to assign values to the variables x_i and z_i such

that the constraints (10)–(12) can be satisfied.

$$\sum_{i \in I} (x_i + \zeta z_i) \leq \eta \quad (12)$$

Furthermore, in the new MIQP formulation, we use the objective function defined in (1) and adopt constraints (2)–(4) from the formulation (M-STPK) of Scozzari et al. [49] to model the budget constraint, the cardinality constraint, and the lower and upper bounds on the portfolio weights of the selected assets.

The new MIQP formulation of the UCITP reads as follows:

$$(M-ST) \begin{cases} \text{Min. (1)} \\ \text{s.t. (2) – (4)} \\ (10) – (12) \\ w_i \geq 0, x_i \geq 0, y_i \in \{0, 1\}, z_i \in \{0, 1\} \quad (i \in I) \end{cases} \quad (13)$$

By modeling the UCITS concentration rule with the constraints (10)–(12), we can reduce the number of constraints in the new MIQP formulation compared to the existing MIQP formulation (M-STPK). Reducing the number of constraints is possible because it is sufficient to use an upper bound on the sum of the weights exceeding the lower UCITS threshold in the MIQP formulation. Hence, we do not have to introduce any constraints that would ensure that the left-hand side of constraint (12) corresponds exactly to the sum of the weights exceeding the lower UCITS threshold. Specifically, the proposed MIQP formulation (M-ST) requires only $4n + 4$ constraints (ignoring those that define the domains of the decision variables), whereas the existing formulation (M-STPK) contains $7n + 4$ constraints.

3.2. New subset representation

A representation is characterized by three components (cf. Gottlieb et al. [24]): the phenotypes, the genotypes, and the decoding procedure that maps the genotypes to the phenotypes.

In the proposed representation of subsets, the phenotypes correspond to subsets of the set of identity tags $I = \{1, \dots, n\}$. As genotype, we use a d -dimensional vector of integers $\mathbf{g} \in \{1, \dots, n\}^d$ with d between \underline{d} and \bar{d} . Here, we assume that the values of the parameters \underline{d} and \bar{d} can be chosen such that the resulting phenotypes are always feasible. The question of how to choose these values for the UCITP is discussed in Section 4.1. The decoding procedure (cf. Algorithm 1) maps a genotype vector to a phenotype S as follows. Each element g_i of the genotype vector is included in the phenotype S if $g_i \notin S$. If $g_i \in S$, then g_i is modified until $g_i \notin S$, and this modified integer g_i is inserted into S . Hence, all phenotypes S correspond to subsets of the set I with a cardinality equal to the dimension d of the corresponding genotype vector. As an example with $n = 6$ and $d = 4$, the vector $[2, 2, 6, 6]$ denotes a possible genotype, which is decoded into the phenotype $\{2, 3, 6, 1\}$.

In the worst case, i.e., if the genotype contains \bar{d} integers that are all identical, then Algorithm 1 requires $\frac{(\bar{d}-1)\bar{d}}{2}$ modifications of the genotype's elements and \bar{d} insertions of the genotype's modified elements into the

Algorithm 1 $\mathcal{O}(\bar{d}^2)$ Decoding

```

1: procedure DEC( $\mathbf{g} \in \{1, \dots, n\}^d$ )
2:    $S := \emptyset$ 
3:   for  $i := 1$  to  $d$  do
4:     while  $g_i \in S$  do
5:        $g_i := (g_i + 1) \bmod n$ 
6:     end while
7:      $S := S \cup \{g_i\}$ 
8:   end for
9:   return  $S$ 
10: end procedure

```

phenotype S . In the best case, i.e., if all elements in the genotype vector are distinct, no element needs to be modified, and Algorithm 1 performs only \bar{d} insertions of the genotype's elements into the phenotype. Hence, the best-case and worst-case time complexities of the decoding procedure are $\mathcal{O}(\bar{d})$ and $\mathcal{O}(\bar{d}^2)$, respectively.

Based on a sorting algorithm that sorts the genotype vector in $\mathcal{O}(\bar{d} \log \bar{d})$ iterations in the worst case (e.g., Quicksort and Mergesort), we design a new decoding procedure (cf. Algorithm 3 in the appendix) that has a better worst-case time complexity than the $\mathcal{O}(\bar{d}^2)$ decoding procedure. The new procedure works as follows. First, the integers in the genotype vector are sorted in a non-decreasing order. Then, all duplicate integers in the genotype are increased (cf. Algorithm 4 in the appendix) such that there are no more duplicate integers in the genotype vector. Since the for-loop in Algorithm 4 is executed $\bar{d} - 1$ times, its worst-case time complexity is $\mathcal{O}(\bar{d})$. The sorted and modified genotype vector is then adjusted such that no integer is larger than n (cf. Algorithm 5 in the appendix). The while-loop in Algorithm 5 is executed \bar{d} times, and thus, the Algorithm 5 has a worst-case time complexity of $\mathcal{O}(\bar{d})$. In total, the worst-case time complexity of Algorithm 3 is therefore $\mathcal{O}(\bar{d} \log \bar{d})$.

In the following, we illustrate the $\mathcal{O}(\bar{d} \log \bar{d})$ decoding procedure by means of a small illustrative example. Suppose for this example that $n = 10$ and that the genotype vector is $[8, 8, 1, 10, 10, 9]$. Then, the decoding procedure is as follows. First, the genotype vector is sorted in a non-decreasing order, which leads to the genotype vector $[1, 8, 8, 9, 10, 10]$. This sorted vector is then modified using Algorithm 4 such that it does not contain any duplicate integers. The result is the genotype vector $[1, 8, 9, 10, 11, 12]$. Finally, using Algorithm 5, the genotype vector is adjusted such that it does not contain any integers larger than n . The resulting decoded phenotype is $\{1, 2, 3, 10, 9, 8\}$.

Two special features of the proposed representation are that the number of all possible genotypes exceeds the number of all possible phenotypes and that not all phenotypes are represented by the same number of genotypes. Hence, the representation exhibits a biased redundancy (cf. Rothlauf [41]). Phenotypes with consecutive identity tags are represented by the most genotypes because the decoding procedure inserts

consecutive identity tags into the phenotypes in place of duplicate integers in the genotypes. This knowledge can be exploited in a very simple way. For example, in mean-variance portfolio-optimization problems, assets with low correlation are likely to be included in an optimal solution. Hence, pairs of weakly correlated assets could be assigned consecutive identity tags.

The properties of feasibility, efficiency, locality, and heritability are investigated in Subsection 4.3.

3.3. Two-stage approach

The two-stage approach proceeds as follows. In the first stage, a genetic algorithm is applied based on the proposed subset representation to obtain a good feasible solution. Then, the local-branching method presented by Fischetti and Lodi [19] is applied based on the new MIQP formulation (M-ST) of the UCITP to improve the solution found in the first stage. In the following, we describe the two stages.

3.3.1. Stage one: genetic algorithm

The genetic algorithm (cf. Algorithm 8 in the appendix) is designed similarly to the simple genetic algorithm described by Rothlauf [41]. First, an initial population is generated at random. Then, the evolutionary process begins and is repeated until a given number of generations n_g is reached. During the evolutionary process, a process of binary tournament selection with replacement (cf. Rothlauf [41]) is applied to determine the mating pool M . The individuals in the mating pool are then either combined with probability p_c using a crossover operator or left unchanged. The resulting individuals are inserted into the population P' . Finally, a mutation operator is applied to the individuals in P' , and the old population P is replaced with the new population P' .

In the genetic algorithm, the fitness of each individual is calculated as follows. The genotype \mathbf{g} is decoded, and the resulting set $\text{DEC}(\mathbf{g})$ of assets is used to define the assets to be included in the portfolio, each with an equal weight of $\frac{1}{|\text{DEC}(\mathbf{g})|}$. Hence, the fitness is calculated using the following function:

$$f(\mathbf{g}) = \frac{1}{|T|} \sum_{t \in T} \left(\sum_{i \in \text{DEC}(\mathbf{g})} r_t^i \frac{1}{|\text{DEC}(\mathbf{g})|} - r_t \right)^2 \quad (14)$$

In the following, we briefly describe the crossover and mutation operators (cf. Algorithms 6 and 7 in the appendix) that are very similar to standard operators from the literature (cf., e.g., Goldberg [23]).

In the mutation operator, a randomly chosen element of the genotype vector is exchanged with a randomly chosen integer from the set $\{1, \dots, n\}$ with probability p_e . In addition to this standard mutation operator, we also allow a randomly chosen element to be added to or removed from the genotype vector with probability p_a or p_r , respectively. Because the dimension of the genotype vector can change during mutation, we must check whether the dimension of the mutated genotype vector is feasible, i.e., whether it is in the range $[\underline{d}, \bar{d}]$. If the mutated genotype vector has a feasible dimension, it is returned; otherwise, the genotype before the mutation is returned.

The crossover operator is very similar to a conventional m -point crossover operator. First, the dimensions of the child genotype vectors \mathbf{g}^3 and \mathbf{g}^4 are set equal to the dimensions of the parent genotype vectors \mathbf{g}^1 and \mathbf{g}^2 , respectively. Then, the crossover point m is randomly chosen. The elements on the left side of m from parent genotype vectors \mathbf{g}^1 and \mathbf{g}^2 are assigned to the child genotype vectors \mathbf{g}^4 and \mathbf{g}^3 , respectively. Furthermore, the remaining elements from parent genotype vectors \mathbf{g}^1 and \mathbf{g}^2 are assigned to the child genotype vectors \mathbf{g}^3 and \mathbf{g}^4 , respectively. With two parent genotype vectors \mathbf{g}^1 and \mathbf{g}^2 that have the same dimension, the operator is identical to the m -point crossover operator. As in the case of the mutation operator, if the dimensions of the input genotype vectors are feasible, then the returned genotype vectors will also have feasible dimensions.

3.3.2. Stage two: local branching

To improve the solution obtained in the first stage, we apply the local-branching method described in Algorithm 2. The algorithm takes as input the genotype vector \mathbf{b} that represents the best individual from stage one. Based on this individual, the MIQP formulation (M-ST-A) is solved to determine an optimal portfolio, i.e., the optimal portfolio weights for the assets selected in the solution from stage one. Then, the set J , which represents the assets selected in the current solution, is initialized. The parameters a and b are also initialized. These parameters are used in the local-branching constraint that is explained below. Then, local branching starts; it is conducted either exactly or heuristically depending on the value used for the parameter n_s . In the following, we describe the exact and heuristic behaviors of the local-branching method.

If n_s is not smaller than the number of index constituents n , then the method operates exactly. In this case, the original set I is used as the set of assets considered during local branching. Then, the MIQP formulation (M-ST-B) is solved. We do not impose a separate time limit for this MIQP in addition to the overall time limit for the two-stage approach. The MIQP formulation (M-ST-B) corresponds to the formulation (M-ST) with the additional local-branching constraint given in (22). This local-branching constraint ensures that at least a and at most b of the binary variables y_i change in value with respect to a previous solution. For this purpose, the first sum in the local-branching constraint counts the number of binary variables y_i , $i \in J$, that had a value of one in the previous solution and have a value of zero in the current solution. The second sum counts the number of binary variables y_i , $i \in I \setminus J$, that had a value of zero in the previous solution and have a value of one in the current solution. If a better solution is found, then the parameters a and b are reset to one and two, respectively. Otherwise, the parameter a is increased to $b + 1$, and b is increased to the new value of a plus 1. Since we do not impose a time limit for solving the MIQP formulation (M-ST-B), we need not consider solutions that could be obtained with a smaller a and b because we know that there is no better solution for a smaller a and b . Hence, if $n_s \geq n$ and no overall time limit for the two-stage approach is imposed, then the local-branching method will eventually find a proven optimal solution.

If we choose $n_s < n$, then the method proceeds heuristically. In this case, only a subset of the set of all index constituents is considered in each iteration of the local-branching method. Specifically, I is set to J , and some randomly selected elements from the set $\{1, \dots, n\}$ are added to I such that the cardinality of I

is equal to n_s . Furthermore, we do not adjust a , and we adjust b only if no better solution could be found by solving the MIQP formulation (M-ST-B). The reason for this is that with $n_s < n$, a better solution with $a = 1$ could exist.

The procedure is repeated until a specified time limit is reached or a proven optimal solution has been found.

$$\begin{aligned}
 \text{(M-ST-A)} \left\{ \begin{aligned} & \text{Min. } \frac{1}{|T|} \sum_{t \in T} \left(\sum_{i \in \text{DEC}(\mathbf{b})} r_t^i w_i - r_t \right)^2 & (15) \\ & \text{s.t. } \sum_{i \in \text{DEC}(\mathbf{b})} w_i = 1 & (16) \\ & \varepsilon \leq w_i \leq \delta & (i \in \text{DEC}(\mathbf{b})) & (17) \\ & w_i - \zeta \leq x_i & (i \in \text{DEC}(\mathbf{b})) & (18) \\ & x_i \leq (\delta - \zeta) z_i & (i \in \text{DEC}(\mathbf{b})) & (19) \\ & \sum_{i \in \text{DEC}(\mathbf{b})} (x_i + \zeta z_i) \leq \eta & (20) \\ & w_i \geq 0, x_i \geq 0, z_i \in \{0, 1\} & (i \in \text{DEC}(\mathbf{b})) & (21) \end{aligned} \right.
 \end{aligned}$$

$$\begin{aligned}
 \text{(M-ST-B)} \left\{ \begin{aligned} & \text{Min. } (1) \\ & \text{s.t. } (2) - (4), (10) - (13) \\ & a \leq \sum_{i \in J} (1 - y_i) + \sum_{i \in I \setminus J} y_i \leq b \end{aligned} \right. \quad (22)
 \end{aligned}$$

4. Numerical experiment

In this section, we report the results of our computational experiment. The objective of this experiment was fourfold. First, we wanted to compare the new subset representation with existing subset representations from the literature. For this purpose, we tested the following solution approaches to the UCITP:

- GA-binary: the genetic algorithm (cf. Algorithm 8) based on a binary representation; for the binary representation, we used the implementation from the genetic algorithm utility library (GAUL; cf. Adcock [1]) with the so-called death penalty for handling infeasible solutions (cf. Moral-Escudero et al. [36]), a bit-exchange operator as the mutation operator, and the m -point crossover operator.
- GA-integer: the genetic algorithm (cf. Algorithm 8) based on an integer representation; for the integer representation, we used the direct subset representation with the Random Respectful Recombination (R3) crossover operator and a bit-exchange mutation operator as described in Moral-Escudero et al. [36].

Algorithm 2 Local Branching – Stage 2

```

1: procedure LOCALBRANCHING(b)
2:    $J := \text{DEC}(\mathbf{b}); a := 1; b := 2;$  Solve (M-ST-A)
3:   while time limit not reached do
4:     if  $n_s \geq n$  then  $I := \{1, \dots, n\}$  else  $I := J;$  Add random elements from  $\{1, \dots, n\}$  to  $I$  until
        $|I| = n_s$  end if
5:     Solve (M-ST-B)
6:     if better solution found then
7:        $a := 1; b := 2; J := \{i \in I : \text{asset } i \text{ is selected in the solution to (M-ST-B)}\}$ 
8:     else
9:       if  $n_s \geq n$  then  $a := b + 1; b := a + 1$  else  $b := b + 1$  end if
10:      if  $a > n$  then return optimal solution end if
11:    end if
12:  end while
13: end procedure

```

- GA- \bar{d}^2 : the genetic algorithm (cf. Algorithm 8) based on the new subset representation with the $\mathcal{O}(\bar{d}^2)$ decoding procedure.
- GA- $\bar{d} \log \bar{d}$: the genetic algorithm (cf. Algorithm 8) based on the new subset representation with the $\mathcal{O}(\bar{d} \log \bar{d})$ decoding procedure and Quicksort as the sorting algorithm.

We ran the genetic algorithm based on the different subset representations for 500 generations, i.e., $n_g = 500$, and report its performance in terms of the four properties feasibility, locality, heritability, and efficiency. To investigate the efficiency, we report the following performance measures:

- OFV: objective function value, i.e., in-sample MSE, of the best feasible portfolio found scaled by a factor of 10^6 . Thereby, a lower OFV is preferred to a higher one.
- TIME: time in seconds to complete the given number of generations. Thereby, a shorter TIME is preferred to a longer one.

Second, we wanted to compare the new MIQP formulation with the existing MIQP formulation from the literature. For this purpose, we tested the following solution approaches to the UCITP:

- M-STPK: an MIQP approach implemented in a commercial mixed-integer programming solver based on the formulation (M-STPK).
- M-ST: an MIQP approach implemented in a commercial mixed-integer programming solver based on the formulation (M-ST).

We ran the MIQP approach based on the two formulations for two time limits of 60 and 120 seconds, and report besides OFV the following in-sample performance measure:

- LB: best lower bound on the objective function value of an optimal portfolio scaled by a factor of 10^6 found within the prescribed time limit. Thereby, a higher LB is preferred to a lower one.

Furthermore, we report the out-of-sample performance measures listed below. Thereby, we let T' be the set of the out-of-sample periods, r_t^P be the portfolio returns in period $t \in T'$, r_t be the index returns in period $t \in T'$, $r_t^D = r_t^P - r_t$ be the differences between the portfolio return and the index return in period $t \in T'$, and $\bar{r}^D = \frac{1}{|T'|} \sum_{t \in T'} r_t^D$ be the average difference between the portfolio returns and the index returns over the out-of-sample periods.

- TE_{RMSE} [%]: tracking error measured by the root-mean-squared error between the portfolio returns and the index returns during the out-of-sample periods, calculated as:

$$\sqrt{\frac{1}{|T'|} \sum_{t \in T'} (r_t^D)^2} \quad (23)$$

Note that the TE_{RMSE} represents the out-of-sample tracking-error measure that corresponds to the in-sample tracking-error measure MSE that is optimized in this paper. Thereby, a lower TE_{RMSE} is preferred to a higher one.

- TE_{TEV} [%]: tracking error measured by the standard deviation of the differences between the portfolio returns and the index returns during the out-of-sample periods calculated as:

$$\sqrt{\frac{1}{|T'|} \sum_{t \in T'} (r_t^D - \bar{r}^D)^2} \quad (24)$$

Note that the TE_{TEV} represents the out-of-sample tracking-error measure that corresponds to the in-sample tracking-error measure TEV that is sometimes used in the literature as an alternative to the MSE. Thereby, a lower TE_{TEV} is preferred to a higher one.

- ER [%]: difference between the cumulative portfolio return and the cumulative index return during the out-of-sample periods, the so-called excess return, calculated as:

$$\prod_{t \in T'} (1 + r_t^P) - \prod_{t \in T'} (1 + r_t) \quad (25)$$

Thereby, the closer the ER to zero, the better the portfolio.

- BETA: slope (beta coefficient) of a regression of the portfolio returns on the index returns. Thereby, the closer the BETA to one, the better the portfolio.
- CORR: correlation between the portfolio returns and the index returns. Thereby, the closer the CORR to one, the better the portfolio.

Third, we wanted to compare the proposed two-stage approach with the genetic algorithm based on the best subset representation and the MIQP approach based on the best MIQP formulation. For this purpose, we tested GA- \bar{d}^2 , M-ST, and the following solution approach:

- TSA: the proposed two-stage approach with the proposed subset representation based on the $\mathcal{O}(\bar{d}^2)$ decoding procedure and a commercial mixed-integer programming solver to solve the MIQP problems within the two-stage approach.

We analyzed the three approaches for two time limits of 120 and 1000 seconds without a limit on the number of generations n_g for GA- \bar{d}^2 . The first stage of the two-stage approach was run for 500 generations, i.e., $n_g = 500$, which took less than 120 seconds for all instances, and the second stage was run for the remaining time. To compare the three approaches, we report the in-sample performance measure OFV and the out-of-sample performance measures TE_{RMSE} , TE_{TEV} , ER, BETA, and CORR.

Fourth, we wanted to investigate the impact of the UCITS concentration rule. For this purpose, we ran TSA with $n_g = 500$ for 120 seconds with and without considering the UCITS concentration rule. Here, we report the in-sample and out-of-sample performance measure OFV and TE_{RMSE} , respectively.

This section is organized as follows. In Subsection 4.1, we explain the test settings used in the experiment. In Subsection 4.2, we present the test instances. In Subsection 4.3, we report the results of the comparison of the subset representations. In Subsection 4.4, we report the results of the comparison of the two MIQP formulations. In Subsection 4.5, we report the results of the comparison of the two-stage approach with the genetic algorithm and the MIQP approach. In Subsection 4.6, we investigate the impact of the UCITS concentration rule.

4.1. Test settings

For the experiments, we used the parameter values listed in Table 4. These parameters were partially given by the problem instances. Some of them, however, could be chosen and would affect the performance of the approaches. For these parameters, we used standard values. To ensure that the cardinality of the phenotypes in the genetic algorithm permitted the construction of a feasible solution, i.e., an equally weighted portfolio satisfying all constraints of the UCITP, we must ensure that $\bar{d} = k$ and $\underline{d} = 20$. Note that we assume here that $\frac{1}{k} \geq \varepsilon$. To see why $\underline{d} = 20$, note that because of the UCITS concentration rule, all weights in an equally weighted portfolio must not exceed a value of 5%, because otherwise the portfolio would be infeasible. Hence, a selection of at least 20 and at most k assets enables the construction of a feasible equally weighted portfolio. Note that, however, when we consider portfolios that are not equally weighted, it is also possible to construct a portfolio that satisfies the UCITS concentration rule with 16 assets, i.e., with weights of 10% assigned to four stocks and weights of 5% assigned to twelve stocks.

For the comparison of the subset representations, slightly different parameter values from those in Table 4 were chosen. We set $l = k = \underline{d} = \bar{d} = 20$ because the subset representation of Moral-Escudero et al. [36] is applicable only to a fixed portfolio cardinality. With a fixed portfolio cardinality, no addition or removal of

Table 4: Values of parameters and sets used in the experiment.

Parameters/sets	Values
$T/T'/n/k/r_t/r_t^i$	depending on the test instance (cf. Subsection 4.2)
$\varepsilon/\zeta/\delta/\eta$	0.01/0.05/0.1/0.4
$l/s/\underline{d}/\bar{d}/n_g/n_s$	16/10n/20/k/500/100
$p_c/p_e/p_a/p_r$	0.5/0.25/0.25/0.25

elements to or from the genotype is possible during mutation. Therefore, we set the probabilities p_r and p_a to zero.

All approaches were implemented in C, and Gurobi 7.0 was used as the mixed-integer programming solver for the MIQP problems. All calculations were performed on an HP Z820 workstation with two 3.1 GHz Intel Xeon CPUs and 128 GB of RAM. All approaches were run five times with the different random seeds zero to four.

4.2. Test instances

We considered 45 problem instances, all derived from real-world data. Each instance comprises the weekly closing prices of n stocks and the index values for 156 weeks. The first 104 weeks were used as the in-sample data for the optimization, and the following 52 weeks were used for out-of-sample evaluations of the portfolios, i.e., $T = \{1, \dots, 104\}$ and $T' = \{105, \dots, 156\}$. For the parameter k , we used the values 20 and 40 except for the Swiss Market Index (SMI) instance and the Hang Seng Index instance, which include only 20 and 31 stocks, respectively.

Instances 1–8 and 24–31 were introduced by Beasley et al. [4] and Canakgoz and Beasley [11] and can be downloaded from the OR-Library (cf. Beasley [3]). Instances 9–23 and 32–45 were introduced by Strub and Baumann [54]. The weekly closing prices and index values for all instances were downloaded with DATASTREAM. Only stocks that were listed in the index during all 156 weeks were included. For each instance, Table 5 lists the number(s) of the instance (corresponding to a specific index and a specific value of k), the name of the index, the number of stocks n , the value(s) of the parameter k , and on which time horizon the data were collected (if known).

4.3. Subset representations

The results in this subsection are reported as averages over all runs and over problem instances 1–23 with the values $l = k = \underline{d} = \bar{d} = 20$, as mentioned above.

First, we investigated the feasibility property of the different representations. The results are summarized in Table 6. The binary representation does not exhibit the feasibility property, as can be seen from the fact that approximately one quarter of all genotypes encountered during the 500 generations represented an infeasible phenotype. For the other representations, all genotypes represented feasible phenotypes. However, the integer representation does not exhibit the feasibility property because special mutation and crossover

Table 5: Problem instances.

Instance no.	Index	n	k	Time horizon
1/24	Hang Seng	31	20/31	1992–1995
2/25	DAX100	85	20/40	1992–1995
3/26	FTSE100	89	20/40	1992–1995
4/27	S&P100	98	20/40	1992–1995
5/28	Nikkei225	225	20/40	1992–1995
6/29	S&P500	457	20/40	NA
7/30	Russell2000	1,319	20/40	NA
8/31	Russell3000	2,152	20/40	NA
9	SMI	20	20	2012–2015
10/32	Hang Seng	49	20/40	2012–2015
11/33	EUROSTOXX50	50	20/40	2012–2015
12/34	FTSE100	96	20/40	2012–2015
13/35	S&P100	99	20/40	2012–2015
14/36	NASDAQ100	101	20/40	2012–2015
15/37	DAX100	102	20/40	2012–2015
16/38	SPI	198	20/40	2012–2015
17/39	Nikkei225	220	20/40	2012–2015
18/40	S&PASX300	254	20/40	2012–2015
19/41	S&P500	489	20/40	2012–2015
20/42	FTSE All Share	567	20/40	2012–2015
21/43	STOXXEURO600	575	20/40	2012–2015
22/44	S&P1200	1,179	20/40	2012–2015
23/45	NASDAQ Composite	2,140	20/40	2012–2015

Table 6: Feasibility: frequencies in [%] of genotypes representing a feasible or an infeasible phenotype.

	GA-binary	GA-integer	GA- \bar{d}^2	GA- $\bar{d} \log \bar{d}$
Feasible	76.97	100.00	100.00	100.00
Infeasible	23.03	0.00	0.00	0.00

operators must be applied and only subsets with a fixed cardinality can be represented. For the proposed representation, however, the feasibility property is satisfied.

Next, we investigated the representations' locality. The locality depends on the mutation operator and on the metric used to measure the distance between phenotypes. We used the mutation operator presented in Subsection 3.3.1. For the distance $d^P(S_1, S_2)$ between two phenotypes S_1 and S_2 , we used the following metric:

$$d^P(S_1, S_2) = \frac{|S_1 \setminus S_2| + |S_2 \setminus S_1| + ||S_1| - |S_2||}{2} \quad (26)$$

This distance metric counts the number of exchanges, removals, or additions necessary to transform one phenotype into the other and can thus be regarded as an edit distance. As an example, for the phenotypes $S_1 = \{1, 2, 3\}$ and $S_2 = \{1, 2, 4\}$, we obtain a distance of $d^P(S_1, S_2) = 1$. Table 7 shows the numbers of mutations for which the distance between the phenotypes before and after mutation was zero, one, and larger than one. From this table, we can gain the following insights:

Table 7: Locality: frequencies in [%] of certain distances $d^P(S, S')$ between the phenotypes before mutation (S) and after mutation (S').

$d^P(S, S')$	GA-binary	GA-integer	GA- \bar{d}^2	GA- $\bar{d} \log \bar{d}$
0	95.14	93.01	75.12	75.12
1	4.86	6.99	24.88	24.88
> 1	0.00	0.00	0.00	0.00

Table 8: Heritability: distance between parent and child phenotypes vs. distance between parent phenotypes. A difference smaller than or equal to zero means that the distance between the parent phenotypes was not smaller than each of the four distances between one of the two parents and one of the two children; a positive difference means that at least one of the four distances between one of the two parents and one of the two children was larger than the distance between the parents. Frequencies are expressed in [%].

	GA-binary	GA-integer	GA- \bar{d}^2	GA- $\bar{d} \log \bar{d}$
≤ 0	100.00	100.00	99.89	99.89
> 0	0.00	0.00	0.11	0.11

- Since there is no case with a distance larger than one, all representations exhibit the locality property.
- The integer and binary representations lead to a smaller fraction of cases with a distance of one. The reason is that the binary and integer representations use a bit-exchange mutation operator that is likely to select two bits with the same value, resulting in a distance of zero.

Furthermore, we analyzed the heritability property. The heritability property is satisfied if after a crossover, the distance between the mother and father phenotypes is no smaller than any of the distances between parent and child phenotypes, i.e., between the phenotypes of mother and daughter, mother and son, father and daughter, and father and son. For each representation, Table 8 lists the frequency of crossovers in which all parent-child distances were no larger than the distance between the parents and the frequency of crossovers in which any parent-child distance was larger than the distance between the parents. From this table, we can conclude that the binary and integer representations perfectly exhibit the heritability property. The proposed representation also exhibits the heritability property, as can be seen from the very small number of cases in which any child-parent distance was larger than the distance between the parents.

Finally, we investigated the efficiency property. From Subsection 3.2, we know that the worst-case time complexity of the decoding procedure for the proposed subset representation is either $\mathcal{O}(\bar{d}^2)$ or $\mathcal{O}(\bar{d} \log \bar{d})$, depending on the algorithm used. These complexities can be regarded as efficient. To see how the efficiency of the proposed representation compares with that of other subset representations, we list in Table 9 the times necessary to complete 500 generations of the genetic algorithm based on the different representations. These times encompass not only the decoding process but also the mutation and crossover operators. From this table, we can see that the proposed representation is the fastest on average. Moreover, the $\mathcal{O}(\bar{d}^2)$ decoding

Table 9: OFV after 500 generations and the corresponding time necessary.

	GA-binary	GA-integer	GA- \bar{d}^2	GA- $\bar{d} \log \bar{d}$
TIME	52.6	33.8	22.5	23.1
OFV	145.3	92.1	88.4	88.4

procedure is slightly faster than the $\mathcal{O}(\bar{d} \log \bar{d})$ decoding procedure, which can be attributed to the best-case
time complexity of $\mathcal{O}(\bar{d})$ for the $\mathcal{O}(\bar{d}^2)$ decoding procedure that is achieved when there are no duplicate
integers in the genotype vectors. This is often the case because \bar{d} is much smaller than n for most of the
instances, which reduces the probability of duplicate integers. Furthermore, Table 9 shows the averages of the
best objective function values after 500 generations. The proposed representation also yields the best results
in terms of the objective function value. The superiority of these results compared with those of the binary
representation can, at least in part, be attributed to the fact that only feasible phenotypes are investigated
when using the proposed representation. Meanwhile, a possible explanation for the superior results compared
with those of the integer representation may be that the higher frequency of actual mutations performed with
the proposed representation (cf. Table 7) was beneficial.

From this first set of experiments, we can conclude that the genetic algorithm based on the novel subset
representation with the $\mathcal{O}(\bar{d}^2)$ decoding procedure yields the best results in terms of TIME and OFV. In
addition, the proposed subset representation exhibits the properties of feasibility, locality, heritability, and
efficiency.

4.4. MIQP formulations

Next, we compared the two formulations (M-STPK) and (M-ST). For all instances, Table 10 lists the
OFV and the LB obtained by M-STPK and M-ST after 60 and 120 seconds. Bold values indicate the better
formulation for each instance and time limit based on the OFV and the LB. As can be seen from this table,
M-STPK was able to devise a feasible portfolio for all instances, whereas M-ST could not devise a feasible
portfolio for two instances within 60 seconds. However, for these two instances, the portfolios determined
by M-STPK within 60 seconds had a very low quality in terms of OFV, which can be seen from the large
improvements that could be made for these two instances in terms of OFV when the approaches were run for
60 more seconds. The main finding from this table is that both formulations lead to similar lower bounds,
but the new formulation yielded better objective function values on average. We also performed two non-
parametric statistical tests, specifically a Wilcoxon signed rank test to compare the median LB obtained by
M-STPK within 120 seconds with that obtained by M-ST within 120 seconds, and another Wilcoxon signed
rank test to compare the median OFV obtained by M-STPK within 120 seconds with that obtained by M-ST
within 120 seconds. These two tests indicated that the median LB are not statistically different at a standard
significance level (p -value: 0.6016) but that the median OFV obtained by M-ST is significantly lower than
that by M-STPK (p -value: 0.0110). Because M-ST seems to perform better in terms of OFV, especially for

the larger instances with $n \geq 457$, we performed another Wilcoxon signed rank test to compare M-ST and M-STPK. Also based on this test, the median OFV obtained after 120 seconds by M-ST is significantly lower than that by M-STPK, even though the p -value increases to 0.0340 due to the smaller sample size.

Furthermore, in Table 11, we compare the out-of-sample performance of the portfolios obtained by M-STPK and M-ST within 120 seconds. As can be seen from this table, all out-of-sample performance measures TE_{RMSE} , TE_{TEV} , ER, BETA, and CORR are almost identical for the two approaches M-STPK and M-ST, even though M-ST was able to devise portfolios with lower objective function values. This finding shows that a lower objective function value does not always lead to a better out-of-sample performance.

For illustrative purposes, we show in Figure 1 the cumulative returns of a selection of S&P indices and the portfolios obtained by M-STPK and M-ST. As can be seen from this figure, the lines that represent the cumulative returns of the portfolios obtained by M-STPK and M-ST look similar to the lines that represent the cumulative returns of the indices. However, the differences in the cumulative returns, i.e., the ER, can become substantial at the end of the out-of-sample periods.

From this second set of experiments, we can conclude that the new MIQP formulation (M-ST) is superior to the existing MIQP formulation (M-STPK) in terms of the OFV obtained within the given time limits. This superiority is the reason why the MIQP formulations used in stage two of the proposed two-stage approach (cf. Subsection 3.3.1) are based on the formulation (M-ST).

4.5. Two-stage approach

In columns four to nine of Table 12, we compare the in-sample performance measure OFV obtained by M-ST, $GA-\bar{d}^2$, and TSA. The results indicate that the proposed two-stage approach yields the best results in terms of the OFV on average for both a short time limit of 120 seconds and a longer time limit of 1000 seconds.

We also evaluated the portfolios constructed using the three approaches over the out-of-sample periods in terms of the TE_{RMSE} . The results are shown in columns 10 to 15 of Table 12. These results indicate that the portfolios' in-sample performances are consistent with the out-of-sample performances in terms of the ranking among the three approaches; i.e., TSA also yielded the best out-of-sample results. However, the longer time limit had no marked influence on the out-of-sample results despite improving the in-sample results.

We also analyzed the differences in the OFV and the TE_{RMSE} using the non-parametric statistical tests implemented in the software package MULTIPLETEST (cf. <http://sci2s.ugr.es/sicidm> and Garcia and Herrera [21]). Tables 13 and 14 report the main results obtained. Table 13 reports the Friedman ranks; a lower rank indicates better performance. According to these ranks, TSA performed best in terms of both the OFV and the TE_{RMSE} . Table 14 reports the p -values obtained using different statistical procedures with respect to the null hypothesis that the performance does not differ between the two approaches represented in each row. For the in-sample period, TSA performed significantly better than both other approaches according to almost all tests. The results look similar for the out-of-sample periods, with the exception that

Table 10: OFV and LB obtained by M-STPK and M-ST within a time limit of 60 and 120 seconds expressed as averages over all runs with different random seeds; bold values indicate the best approach in terms of the OFV or LB for each instance and time limit.

Instance			OFV				LB			
			60 seconds		120 seconds		60 seconds		120 seconds	
			M-STPK	M-ST	M-STPK	M-ST	M-STPK	M-ST	M-STPK	M-ST
n	k	No.								
20	20	9	90.9	90.9	90.9	90.9	90.9	90.9	90.9	90.9
31	20	1	56.1	56.1	56.1	56.1	56.1	56.1	56.1	56.1
31	31	24	47.3	47.3	47.3	47.3	47.3	47.3	47.3	47.3
49	20	10	27.8	27.8	27.8	27.8	27.5	26.4	27.8	27.8
49	40	32	13.3	13.3	13.3	13.3	13.3	13.3	13.3	13.3
50	20	11	24.0	23.6	24.0	23.6	12.6	12.7	14.1	14.2
50	40	33	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9
85	20	2	25.0	25.2	24.6	24.6	5.3	5.3	6.2	6.4
85	40	25	3.7	3.7	3.7	3.7	3.7	3.7	3.7	3.7
89	20	3	62.2	59.0	61.8	58.6	16.7	17.2	18.5	18.9
89	40	26	12.8	12.9	12.7	12.8	10.2	10.1	10.5	10.5
96	20	12	22.6	23.8	22.3	23.3	5.5	5.1	6.3	6.1
96	40	34	4.0	4.1	3.9	4.0	3.0	3.0	3.1	3.1
98	20	4	36.8	38.0	36.8	37.8	7.5	7.1	8.6	8.5
98	40	27	6.2	6.4	6.1	6.2	4.6	4.7	4.8	4.9
99	20	13	23.3	22.2	22.4	21.0	2.7	2.6	3.2	3.2
99	40	35	3.9	3.8	3.9	3.7	1.2	1.3	1.3	1.3
101	20	14	48.7	46.7	48.7	46.7	23.4	22.4	25.5	24.7
101	40	36	16.6	16.6	16.6	16.6	16.1	16.0	16.4	16.3
102	20	15	28.4	27.7	28.4	27.7	12.9	12.0	13.9	13.4
102	40	37	9.9	9.9	9.9	9.9	9.9	9.9	9.9	9.9
198	20	16	35.6	34.7	35.0	34.6	23.6	23.6	24.6	24.6
198	40	38	22.0	22.0	22.0	22.0	22.0	22.0	22.0	22.0
220	20	17	71.9	56.4	45.7	49.5	0.0	0.0	0.0	0.0
220	40	39	12.1	9.7	7.5	7.5	0.0	0.0	0.0	0.0
225	20	5	96.1	97.8	80.2	73.5	0.0	0.0	0.0	0.0
225	40	28	14.2	26.3	9.6	9.6	0.0	0.0	0.0	0.0
254	20	18	55.0	60.2	34.4	37.1	0.0	0.0	0.0	0.0
254	40	40	11.5	17.2	5.7	4.9	0.0	0.0	0.0	0.0
457	20	6	258.8	143.2	155.0	118.7	0.0	0.0	0.0	0.0
457	40	29	131.6	49.1	63.0	20.5	0.0	0.0	0.0	0.0
489	20	19	126.4	58.2	112.4	45.3	0.0	0.0	0.0	0.0
489	40	41	36.3	24.5	27.8	9.9	0.0	0.0	0.0	0.0
567	20	20	80.9	50.5	80.9	44.2	0.0	0.0	0.0	0.0
567	40	42	56.0	204.9	56.0	201.5	0.0	0.0	0.0	0.0
575	20	21	164.9	70.9	164.9	70.9	0.0	0.0	0.0	0.0
575	40	43	61.1	28.3	61.0	27.5	0.0	0.0	0.0	0.0
1179	20	22	252.5	87.7	251.8	87.7	0.0	0.0	0.0	0.0
1179	40	44	61.9	27.6	60.7	27.6	0.0	0.0	0.0	0.0
1319	20	7	430.3	447.8	401.2	447.8	0.0	0.0	0.0	0.0
1319	40	30	260.8	163.7	246.1	163.7	0.0	0.0	0.0	0.0
2140	20	23	1 703.9	–	533.5	250.1	0.0	–	0.0	0.0
2140	40	45	745.2	152.7	732.9	119.4	0.0	0.0	0.0	0.0
2152	20	8	3 346.9	–	204.5	240.7	0.0	–	0.0	0.0
2152	40	31	108.3	72.8	70.3	58.7	0.0	0.0	0.0	0.0
Average			85.2	57.5	89.5	60.8	9.8	9.7	9.6	9.6

The symbol “–” indicates that no feasible portfolio was found within the prescribed computational time limit.

Averages are computed over the instances for which both M-STPK and M-ST found a feasible portfolio.

Figure 1: Cumulative returns of different S&P indices and portfolios obtained by M-STPK and M-ST within 120 seconds with a random seed of zero and a value of 20 for k .

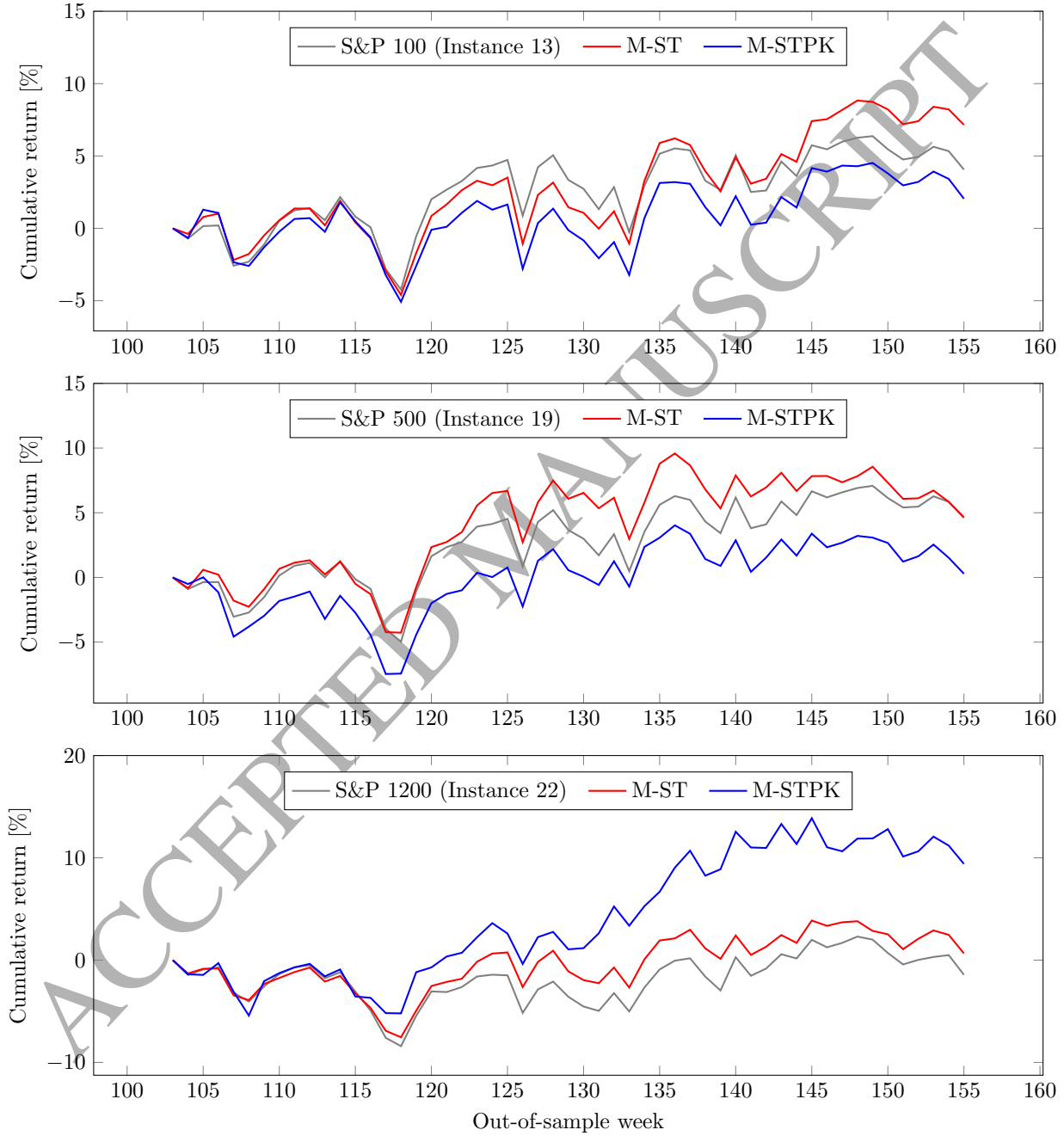


Table 11: Out-of-sample performance of the portfolios obtained by M-STPK and M-ST within 120 seconds in terms of TE_{RMSE} , TE_{TEV} , ER, BETA, and CORR expressed as averages over all instances and runs with different random seeds.

Approach	TE_{RMSE}	TE_{TEV}	ER	BETA	CORR
M-STPK	0.50	0.49	0.35	0.97	0.96
M-ST	0.49	0.48	0.30	0.97	0.96

the performance difference between M-ST and TSA with a time limit of 120 seconds was not statistically significant.

To further investigate the out-of-sample performance of the portfolios obtained by TSA, we compare in Table 15 the out-of-sample performance measures TE_{TEV} , ER, BETA, and CORR of the portfolios obtained with TSA with those obtained with M-ST. We report the results for a time limit of 120 seconds only, because as shown in Table 12, the longer time limit had no marked influence on the out-of-sample performance in terms of the TE_{RMSE} . Also, we do not show the results of $GA-\bar{d}^2$ in this table, because as shown before, $GA-\bar{d}^2$ lead to an inferior in-sample and out-of-sample performance than M-ST and TSA. The main finding from Table 15 is that TSA yielded slightly better portfolios than M-ST in terms of all out-of-sample performance measures.

To further investigate the performance of TSA compared to M-ST, we compare in Table 16 the in-sample performance measure OFV and the out-of-sample performance measures TE_{RMSE} , TE_{TEV} , ER, BETA, and CORR for a selection of S&P indices and different values for the maximum portfolio cardinality k . As Table 16 shows, TSA performed better on average in terms of all performance measures except of the ER. Specifically, TSA lead to a larger positive ER than M-ST did, which is in the considered case of index tracking inferior to a lower positive one. Also, Table 16 shows that larger values for k tend to lead to better index-tracking portfolios in terms of the different out-of-sample performance measures.

For illustrative purposes, we again show in Figure 2 the cumulative returns of a selection of S&P indices and the portfolios obtained by M-ST and TSA. A comparison of Figure 2, in which instances with $k = 40$ are considered, with Figure 1, in which instances with $k = 20$ are considered, indicates that larger values for the maximum portfolio cardinality k allow the construction of better index-tracking portfolios.

From this third set of experiments, we can conclude that TSA is superior to $GA-\bar{d}^2$ and M-ST in terms of both the in-sample and the out-of-sample performance.

4.6. Impact of UCITS concentration rule

Table 17 shows for all instances the results of TSA when the UCITS concentration rule was ignored. Columns two to four show the results for the instances with $k = 20$. The results of the remaining instances are shown in columns five to seven. From Table 17, we can gain the following main insights for the instances with $k = 20$:

- Ignoring the concentration rule allowed the determination of portfolios that had a lower OFV, but

Table 12: OFV and corresponding TE_{RMSE} obtained by M-ST, $GA-\bar{d}^2$, and TSA within a time limit of 120 and 1000 seconds expressed as averages over all runs with different random seeds; bold values indicate the best approach in terms of the OFV and the TE_{RMSE} for each instance and time limit.

Instance			OFV						TE_{RMSE}					
			120 seconds			1000 seconds			120 seconds			1000 seconds		
			M-ST	$GA-\bar{d}^2$	TSA	M-ST	$GA-\bar{d}^2$	TSA	M-ST	$GA-\bar{d}^2$	TSA	M-ST	$GA-\bar{d}^2$	TSA
20	20	9	90.9	311.3	90.9	90.9	311.3	90.9	0.34	0.58	0.34	0.34	0.58	0.34
31	20	1	56.1	194.2	56.1	56.1	194.2	56.1	0.47	0.69	0.47	0.47	0.69	0.47
31	31	24	47.3	194.2	47.3	47.3	194.2	47.3	0.46	0.69	0.46	0.46	0.69	0.46
49	20	10	27.8	62.0	27.8	27.8	61.5	27.8	0.32	0.43	0.32	0.32	0.42	0.32
49	40	32	13.3	60.8	13.3	13.3	60.8	13.3	0.21	0.40	0.21	0.21	0.40	0.21
50	20	11	23.6	48.8	24.4	23.6	47.8	24.0	0.26	0.26	0.28	0.26	0.26	0.27
50	40	33	5.9	25.2	5.9	5.9	25.3	5.9	0.14	0.20	0.14	0.14	0.20	0.14
85	20	2	24.6	54.2	24.1	23.9	52.4	23.9	0.34	0.36	0.32	0.35	0.37	0.32
85	40	25	3.7	46.7	3.8	3.7	47.5	3.7	0.22	0.35	0.23	0.22	0.35	0.22
89	20	3	58.6	98.9	68.7	56.4	97.4	65.7	0.38	0.38	0.43	0.40	0.38	0.44
89	40	26	12.8	43.4	13.1	12.4	44.5	12.4	0.26	0.28	0.23	0.27	0.27	0.27
96	20	12	23.3	47.8	23.3	22.5	46.7	23.1	0.31	0.33	0.30	0.31	0.31	0.30
96	40	34	4.0	32.6	3.9	3.9	33.1	3.9	0.19	0.28	0.19	0.20	0.28	0.18
98	20	4	37.8	75.2	43.7	34.8	73.2	35.3	0.40	0.41	0.40	0.41	0.42	0.41
98	40	27	6.2	38.9	7.1	5.9	38.4	5.9	0.28	0.33	0.24	0.29	0.33	0.29
99	20	13	21.0	39.5	21.3	19.9	39.6	21.3	0.40	0.37	0.35	0.40	0.37	0.35
99	40	35	3.7	17.2	3.7	3.6	16.2	3.6	0.25	0.28	0.26	0.27	0.28	0.26
101	20	14	46.7	128.0	47.0	45.6	127.4	46.7	0.48	0.61	0.47	0.53	0.58	0.48
101	40	36	16.6	126.5	16.6	16.6	123.3	16.6	0.35	0.59	0.35	0.35	0.59	0.35
102	20	15	27.7	62.1	27.7	27.6	63.8	27.7	0.29	0.34	0.29	0.29	0.35	0.29
102	40	37	9.9	54.8	9.9	9.9	56.8	9.9	0.20	0.32	0.20	0.20	0.33	0.20
198	20	16	34.6	116.4	34.9	34.4	119.1	34.4	0.30	0.50	0.30	0.30	0.52	0.30
198	40	38	22.0	113.3	22.0	22.0	111.6	22.0	0.35	0.43	0.34	0.35	0.41	0.35
220	20	17	49.5	78.6	39.2	32.7	80.2	38.6	0.54	0.51	0.55	0.52	0.49	0.55
220	40	39	7.5	35.5	4.8	5.5	36.1	4.5	0.43	0.41	0.46	0.39	0.39	0.42
225	20	5	73.5	84.0	48.9	45.9	87.6	45.0	0.47	0.48	0.54	0.45	0.50	0.54
225	40	28	9.6	23.0	6.4	6.0	21.5	6.1	0.36	0.34	0.36	0.35	0.34	0.36
254	20	18	37.1	69.8	21.5	19.3	67.6	20.1	0.31	0.44	0.33	0.34	0.39	0.33
254	40	40	4.9	55.7	3.8	2.7	54.6	3.1	0.23	0.43	0.26	0.22	0.50	0.23
457	20	6	118.7	95.5	58.9	77.2	93.9	54.6	0.89	0.86	0.82	0.93	0.88	0.85
457	40	29	20.5	28.0	6.6	12.2	23.3	6.4	0.68	0.70	0.67	0.61	0.69	0.67
489	20	19	45.3	33.8	22.8	28.7	36.0	21.4	0.45	0.50	0.48	0.50	0.54	0.48
489	40	41	9.9	10.0	2.9	5.9	8.1	2.3	0.35	0.36	0.34	0.34	0.36	0.34
567	20	20	44.2	43.6	22.6	24.6	42.7	20.8	0.33	0.41	0.36	0.36	0.40	0.35
567	40	42	201.5	15.2	2.8	4.8	13.1	2.3	0.42	0.30	0.28	0.26	0.30	0.28
575	20	21	70.9	39.5	24.5	43.5	38.7	22.9	0.48	0.52	0.48	0.49	0.52	0.48
575	40	43	27.5	15.4	3.1	6.6	11.3	3.0	0.34	0.35	0.36	0.41	0.31	0.36
1179	20	22	87.7	35.0	23.9	63.9	34.4	23.3	0.55	0.48	0.47	0.50	0.48	0.47
1179	40	44	27.6	16.1	6.9	11.2	8.6	2.0	0.49	0.39	0.40	0.49	0.38	0.44
1319	20	7	447.8	156.7	115.2	295.5	154.7	113.6	1.93	1.56	1.45	2.19	1.54	1.47
1319	40	30	163.7	64.6	64.6	65.2	43.0	11.1	1.48	1.25	1.25	1.71	1.19	1.28
2140	20	23	250.1	65.6	49.0	241.3	65.9	45.8	0.91	0.60	0.57	0.87	0.61	0.61
2140	40	45	119.4	62.0	62.0	100.5	27.9	5.1	0.84	0.68	0.68	0.85	0.62	0.59
2152	20	8	240.7	78.6	58.3	269.6	78.2	53.6	1.16	0.94	0.95	1.19	1.00	0.95
2152	40	31	58.7	63.6	63.6	56.6	27.8	5.5	1.17	0.67	0.67	1.13	0.72	0.75
Average			60.8	70.3	30.0	45.0	67.6	25.2	0.49	0.50	0.44	0.50	0.50	0.45

Figure 2: Cumulative returns of different S&P indices and portfolios obtained by M-ST and TSA within 120 seconds with a random seed of zero and a value of 40 for k .

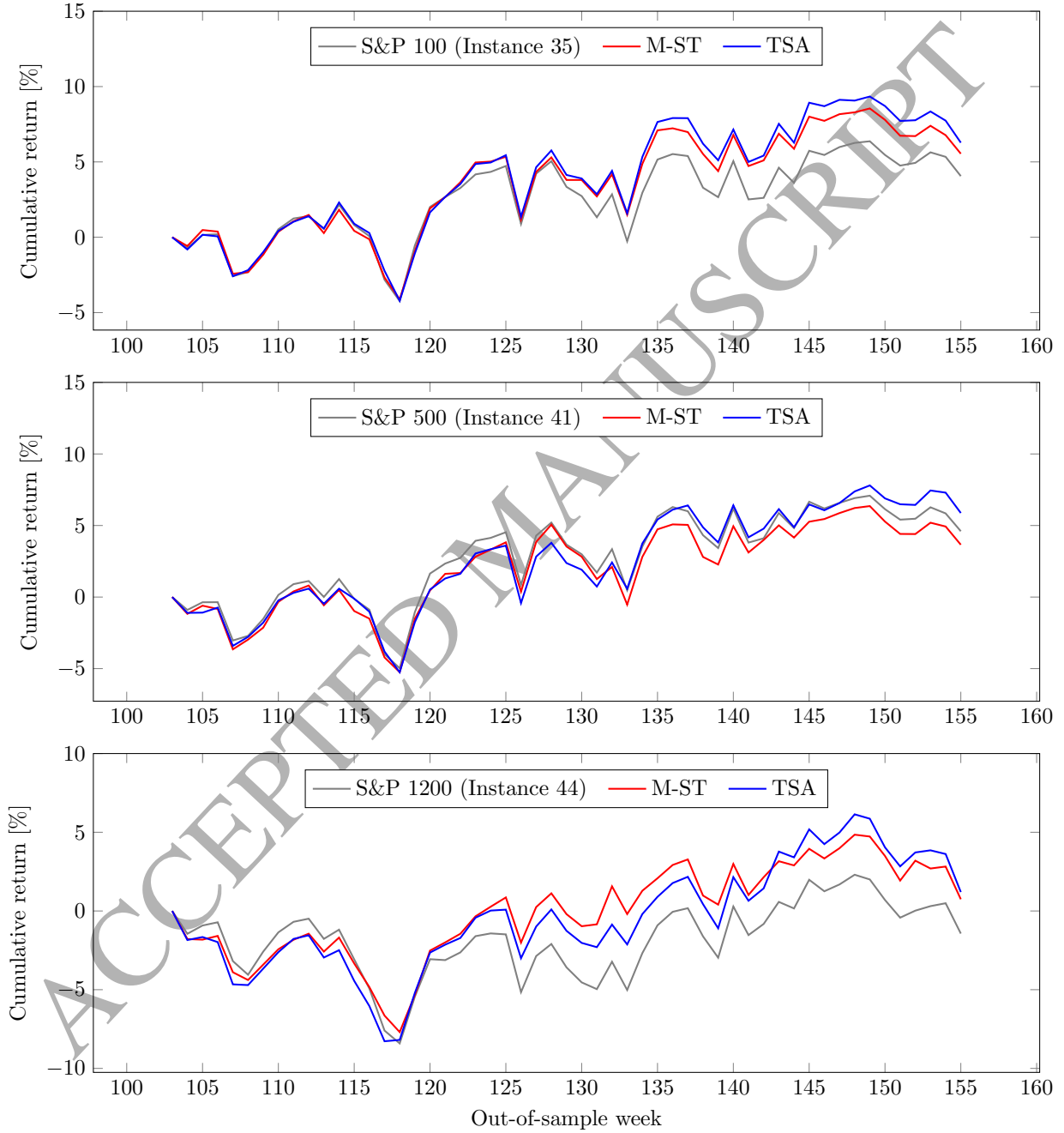


Table 13: Friedman ranks.

	in-sample		out-of-sample	
	120 s	1000 s	120 s	1000 s
M-ST	1.93	1.78	1.91	1.98
GA- \vec{d}^2	2.68	2.80	2.43	2.40
TSA	1.39	1.42	1.66	1.62

Table 14: p -values of multiple comparisons between all algorithms. Cases in which the null hypothesis can be rejected at a significance level of $\alpha = 0.1$ are marked in bold. Post hoc procedures: Nemenyi, Holm, Shaffer, Berg.

	Hypothesis	in-sample				out-of-sample			
		Neme	Holm	Shaf	Berg	Neme	Holm	Shaf	Berg
120 s	GA- \vec{d}^2 vs. TSA	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	M-ST vs. GA- \vec{d}^2	0.00	0.00	0.00	0.00	0.04	0.03	0.01	0.01
	M-ST vs. TSA	0.03	0.01	0.01	0.01	0.68	0.23	0.23	0.23
1000 s	GA- \vec{d}^2 vs. TSA	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	M-ST vs. GA- \vec{d}^2	0.00	0.00	0.00	0.00	0.14	0.09	0.05	0.05
	M-ST vs. TSA	0.28	0.09	0.09	0.09	0.28	0.09	0.09	0.09

violated the concentration rule. Hence, the concentration rule seems to be binding when small values for k are considered.

- The TE_{RMSE} increased when the concentration rule was ignored. Hence, we can argue that the concentration rule reduces the portfolio risk in terms of the out-of-sample risk relative to the index. Recall, however, that we consider the construction of a new portfolio from cash in this paper. Hence, further investigations would be required to investigate the influence of the concentration rule on the out-of-sample risk when a periodic rebalancing is considered.

For the instances with a larger value of k , the impact of the concentration rule on the results can be neglected, because the average portfolio weights decrease with larger values of k , and portfolio weights larger than the UCITS lower concentration-rule threshold occur less often.

From this fourth set of experiments, we can conclude that the UCITS regulations are binding for small values of k and reduce the out-of-sample risk of the portfolio relative to the index.

5. Conclusion

We presented a two-stage approach to the UCITS-constrained index-tracking problem based on a genetic algorithm and a local-branching method. For the genetic algorithm, we presented a new representation of subsets, and for the local-branching method, we presented a novel MIQP formulation of the UCITP. We

Table 15: Out-of-sample performance of M-ST and TSA after 120 seconds in terms of TE_{TEV} , ER, BETA, and CORR expressed as averages over all runs with different random seeds.

n	k	No.	TE_{TEV}		ER		BETA		CORR	
			M-ST	TSA	M-ST	TSA	M-ST	TSA	M-ST	TSA
20	20	9	0.33	0.33	-3.57	-3.57	1.02	1.02	0.99	0.99
31	20	1	0.45	0.45	4.22	4.22	0.99	0.99	0.99	0.99
31	31	24	0.45	0.45	3.79	3.79	1.00	1.00	0.99	0.99
49	20	10	0.32	0.32	2.54	2.54	1.02	1.02	0.99	0.99
49	40	32	0.21	0.21	2.34	2.34	1.01	1.01	0.99	0.99
50	20	11	0.26	0.28	1.15	0.44	1.00	1.01	1.00	0.99
50	40	33	0.13	0.13	1.82	1.82	1.01	1.01	1.00	1.00
85	20	2	0.34	0.32	-1.87	-2.15	0.95	0.96	0.99	0.99
85	40	25	0.22	0.23	1.14	0.72	0.96	0.96	0.99	0.99
89	20	3	0.38	0.43	0.61	-0.11	0.97	0.95	0.98	0.98
89	40	26	0.26	0.23	0.08	0.38	0.98	0.99	0.99	0.99
96	20	12	0.31	0.30	0.26	1.40	0.95	0.96	0.99	0.99
96	40	34	0.19	0.18	-1.42	-1.02	0.98	0.99	1.00	1.00
98	20	4	0.40	0.40	1.66	0.04	0.97	0.96	0.96	0.96
98	40	27	0.28	0.24	2.43	0.20	1.07	1.03	0.98	0.99
99	20	13	0.40	0.35	1.73	-0.47	0.96	0.95	0.97	0.98
99	40	35	0.24	0.26	0.94	0.71	1.01	1.01	0.99	0.99
101	20	14	0.48	0.47	-2.32	-2.23	1.01	1.02	0.97	0.97
101	40	36	0.35	0.35	1.87	1.97	1.04	1.03	0.98	0.98
102	20	15	0.29	0.29	-0.23	-0.14	1.01	1.00	0.99	0.99
102	40	37	0.20	0.20	-2.23	-2.23	1.00	1.00	1.00	1.00
198	20	16	0.30	0.29	-2.44	-2.44	1.06	1.06	0.99	0.99
198	40	38	0.35	0.33	-2.79	-2.99	1.03	1.04	0.99	0.99
220	20	17	0.52	0.54	0.67	-1.06	0.98	0.95	0.98	0.97
220	40	39	0.43	0.45	3.79	3.35	1.02	0.97	0.98	0.98
225	20	5	0.47	0.54	1.34	1.03	0.92	0.96	0.97	0.96
225	40	28	0.36	0.35	1.50	1.54	0.96	0.94	0.98	0.98
254	20	18	0.31	0.33	-0.28	-1.54	0.96	0.95	0.99	0.98
254	40	40	0.22	0.26	-0.62	-1.93	0.99	0.98	0.99	0.99
457	20	6	0.88	0.81	-6.57	-4.43	0.95	0.98	0.93	0.94
457	40	29	0.67	0.66	2.61	-0.56	1.00	1.01	0.96	0.96
489	20	19	0.45	0.48	-0.53	1.45	0.98	0.97	0.96	0.95
489	40	41	0.35	0.34	0.70	1.18	0.98	1.01	0.98	0.98
567	20	20	0.33	0.36	1.50	1.34	0.98	0.99	0.98	0.98
567	40	42	0.42	0.28	1.16	0.58	0.98	0.99	0.96	0.99
575	20	21	0.48	0.48	2.46	1.84	1.00	0.98	0.98	0.98
575	40	43	0.33	0.36	-0.78	2.44	0.99	1.00	0.99	0.99
1179	20	22	0.54	0.47	-3.29	0.92	0.91	0.94	0.95	0.96
1179	40	44	0.48	0.40	2.12	0.11	0.99	0.98	0.95	0.97
1319	20	7	1.91	1.44	1.26	5.18	0.61	0.76	0.71	0.82
1319	40	30	1.47	1.23	4.56	-10.13	0.77	0.84	0.80	0.86
2140	20	23	0.91	0.57	-3.25	-2.71	0.95	0.95	0.86	0.95
2140	40	45	0.82	0.66	-4.56	-7.98	0.96	0.96	0.89	0.93
2152	20	8	1.14	0.95	-5.72	-0.36	0.85	0.98	0.89	0.91
2152	40	31	1.16	0.66	5.87	-3.65	0.90	1.02	0.87	0.96
Average			0.48	0.44	0.30	-0.23	0.97	0.98	0.96	0.97

Table 16: In-sample and out-of-sample performance of M-ST and TSA for different values of k after 120 seconds in terms of OFV, TE_{RMSE} , TE_{TEV} , ER, BETA, and CORR expressed as averages over all runs with different random seeds for the instances introduced by Strub and Baumann [54] that are based on the S&P indices.

	k	OFV		TE_{RMSE}		TE_{TEV}		ER		BETA		CORR	
		M-ST	TSA	M-ST	TSA	M-ST	TSA	M-ST	TSA	M-ST	TSA	M-ST	TSA
S&P 100	20	21.0	21.3	0.40	0.35	0.40	0.35	1.73	-0.47	0.96	0.95	0.97	0.98
	30	8.9	8.8	0.30	0.31	0.30	0.30	-0.29	-0.56	1.01	0.98	0.98	0.98
	40	3.7	3.7	0.25	0.26	0.24	0.26	0.94	0.71	1.01	1.01	0.99	0.99
	50	1.9	1.9	0.22	0.21	0.21	0.20	2.39	1.21	1.01	1.01	0.99	0.99
	60	1.3	1.3	0.19	0.19	0.18	0.18	1.79	1.79	1.02	1.02	0.99	0.99
	70	1.3	1.3	0.19	0.19	0.18	0.18	1.79	1.79	1.02	1.02	0.99	0.99
	80	1.3	1.3	0.19	0.19	0.18	0.18	1.79	1.79	1.02	1.02	0.99	0.99
	90	1.3	1.3	0.19	0.19	0.18	0.18	1.79	1.79	1.02	1.02	0.99	0.99
S&P 500	20	45.3	22.8	0.45	0.48	0.45	0.48	-0.53	1.45	0.98	0.97	0.96	0.95
	30	19.6	6.3	0.36	0.34	0.35	0.34	-1.89	0.82	0.97	1.02	0.98	0.98
	40	9.9	2.9	0.35	0.34	0.35	0.34	0.70	1.18	0.98	1.01	0.98	0.98
	50	5.3	1.3	0.30	0.30	0.30	0.30	0.69	-0.61	1.00	0.99	0.98	0.98
	60	5.1	0.7	0.29	0.28	0.29	0.28	0.55	1.15	0.99	1.02	0.98	0.99
	70	5.4	0.5	0.28	0.25	0.27	0.25	0.76	-0.09	1.01	1.03	0.99	0.99
	80	1.1	0.6	0.25	0.26	0.25	0.26	1.02	0.35	0.99	1.03	0.99	0.99
	90	2.5	0.8	0.25	0.27	0.25	0.26	-0.67	-0.14	1.00	1.02	0.99	0.99
S&P 1200	20	87.7	23.9	0.55	0.47	0.54	0.47	-3.29	0.92	0.91	0.94	0.95	0.96
	30	44.6	7.5	0.48	0.42	0.48	0.42	-1.14	0.36	0.92	0.98	0.95	0.96
	40	27.6	6.9	0.49	0.40	0.48	0.40	2.12	0.11	0.99	0.98	0.95	0.97
	50	8.8	14.6	0.37	0.31	0.37	0.31	-1.43	0.29	0.99	0.96	0.97	0.98
	60	4.4	12.7	0.38	0.34	0.37	0.34	-0.54	0.75	1.01	0.98	0.97	0.98
	70	3.9	11.3	0.37	0.30	0.36	0.30	-0.88	1.56	1.00	0.97	0.97	0.98
	80	3.8	10.1	0.37	0.29	0.37	0.29	-0.52	1.76	0.99	1.01	0.97	0.98
	90	4.5	9.9	0.37	0.26	0.37	0.26	-0.69	-0.93	0.99	0.98	0.97	0.99
Average		13.3	7.2	0.33	0.30	0.32	0.30	0.26	0.75	0.99	1.00	0.98	0.98

Table 17: Impact of the UCITS concentration rule; $\sum_{i \in I: w_i > 0.05} w_i$: sum of weights in the best portfolio that exceed the UCITS lower threshold; DIFF OFV and DIFF TE_{RMSE} : difference of OFV and TE_{RMSE} (in percentage points), respectively, between TSA when the concentration rule is ignored and TSA when the concentration rule is considered; values expressed as averages over all runs with different random seeds; negative values indicate lower values for the case when the concentration rule is ignored; time limit: 120 seconds.

n	$k = 20$			$k > 20$		
	$\sum_{i \in I: w_i > 0.05} w_i$	DIFF OFV	DIFF TE_{RMSE}	$\sum_{i \in I: w_i > 0.05} w_i$	DIFF OFV	DIFF TE_{RMSE}
20	0.69	-4.25	0.33	-	-	-
31	0.69	-8.38	0.47	0.65	-2.91	0.16
49	0.64	-1.63	0.06	0.31	0.00	0.00
50	0.62	-0.76	0.00	0.19	0.00	0.00
85	0.64	-2.81	0.09	0.45	-0.17	-0.05
89	0.60	-3.59	-0.01	0.16	0.00	0.00
96	0.58	-0.55	0.05	0.36	0.00	0.00
98	0.57	-2.47	0.16	0.21	0.00	0.00
99	0.60	-1.23	0.16	0.11	0.00	-0.01
101	0.58	-1.33	0.59	0.26	0.00	0.00
102	0.66	-2.15	-0.08	0.30	0.00	0.00
198	0.58	-1.06	0.01	0.33	0.00	0.00
220	0.57	0.97	0.18	0.22	-0.10	-0.06
225	0.57	0.67	-0.03	0.05	-0.12	-0.03
254	0.65	-1.61	0.14	0.39	-0.18	-0.04
457	0.56	-1.51	0.21	0.12	0.00	0.00
489	0.58	-2.17	0.02	0.04	0.00	0.00
567	0.61	-2.57	0.11	0.18	0.00	0.00
575	0.62	1.65	0.26	0.10	0.00	0.00
1179	0.59	-2.30	-0.03	0.01	-0.62	0.04
1319	0.62	-6.73	-0.09	0.00	0.00	0.00
2140	0.67	-2.68	0.47	0.00	0.00	0.00
2152	0.61	-1.71	0.02	0.00	0.00	0.00
Average	0.61	-2.10	0.13	0.20	-0.19	0.00

tested the proposed two-stage approach in a computational experiment based on real-world data. The results demonstrate that the proposed two-stage approach yields significantly better results than either a genetic algorithm or an MIQP approach within a set of given time limits.

Future research should investigate whether the two-stage approach's performance can be improved by exploiting the biased redundancy of the new subset representation or by using a parameter-tuning approach such as that presented in López-Ibáñez et al. [32]. Furthermore, additional practical portfolio constraints could be considered, such as those presented by Filippi et al. [18], Guastaroba and Speranza [25] and Strub and Baumann [54]. A further promising direction for future research would be to investigate the performance of genetic algorithms based on the new subset representation for other optimization problems that involve the selection of a subset, such as the feature-selection problem in machine learning.

Appendix A. Further algorithms

Algorithm 3 $\mathcal{O}(\bar{d} \log \bar{d})$ Decoding

```

1: procedure FASTDEC( $\mathbf{g} \in \{1, \dots, n\}^d$ )
2:   Sort  $\mathbf{g}$  in non-decreasing order
3:    $\mathbf{g} := \text{REMOVEDUPLICATES}(\mathbf{g})$ 
4:    $S := \text{ADJUST}(\mathbf{g})$ 
5:   return  $S$ 
6: end procedure

```

Algorithm 4 $\mathcal{O}(\bar{d} \log \bar{d})$ Decoding – REMOVEDUPLICATES procedure

```

1: procedure REMOVEDUPLICATES( $\mathbf{g} \in \{1, \dots, n\}^d$ )
2:   for  $i := 2$  to  $d$  do
3:     if  $g_{i-1} \geq g_i$  then
4:        $g_i := g_{i-1} + 1$ 
5:     end if
6:   end for
7:   return  $\mathbf{g}$ 
8: end procedure

```

Algorithm 5 $\mathcal{O}(\bar{d} \log \bar{d})$ Decoding – ADJUST procedure

```

1: procedure ADJUST( $\mathbf{g} \in \{1, \dots, n\}^d$ )
2:    $S := \emptyset$ ;  $i := 1$ ;  $m := 1$ ;  $j := d$ 
3:   while  $j \geq i$  do
4:     if  $g_j > n$  then
5:       if  $g_i = m$  then
6:          $S := S \cup \{g_i\}$ ;  $i := i + 1$ ;  $m := m + 1$ 
7:       else
8:          $S := S \cup \{m\}$ ;  $j := j - 1$ ;  $m := m + 1$ 
9:       end if
10:    else
11:       $S := S \cup \{g_j\}$ ;  $j := j - 1$ 
12:    end if
13:  end while
14:  return  $S$ 
15: end procedure

```

Algorithm 6 Mutation

```

1: procedure MUTATE( $\mathbf{g}^1 \in \{1, \dots, n\}^{d_1}$ )
2:    $\mathbf{g}^2 := \mathbf{g}^1$ ;  $d_2 := d_1$ 
3:   if  $\text{random} < p_e$  then
4:     Randomly choose  $i \in \{1, \dots, d_2\}$ ,  $j \in \{1, \dots, n\}$ ;  $g_i^2 := j$ 
5:   end if
6:   if  $\text{random} < p_r$  then
7:     Randomly choose  $i \in \{1, \dots, d_2\}$ ; Remove element  $g_i^2$  from  $\mathbf{g}^2$ ;  $d_2 := d_2 - 1$ 
8:   end if
9:   if  $\text{random} < p_a$  then
10:    Randomly choose  $j \in \{1, \dots, n\}$ ; Add  $j$  to  $\mathbf{g}^2$ ;  $d_2 := d_2 + 1$ 
11:   end if
12:   if  $\underline{d} \leq d_2 \leq \bar{d}$  then return  $\mathbf{g}^2$  else return  $\mathbf{g}^1$  end if
13: end procedure

```

Algorithm 7 Crossover

```

1: procedure CROSSOVER( $\mathbf{g}^1 \in \{1, \dots, n\}^{d_1}, \mathbf{g}^2 \in \{1, \dots, n\}^{d_2}$ )
2:    $d_3 := d_1; d_4 := d_2$ 
3:   Initialize  $\mathbf{g}^3 \in \{1, \dots, n\}^{d_3}, \mathbf{g}^4 \in \{1, \dots, n\}^{d_4}$ ; Randomly choose  $m \in \{1, \dots, \min\{d_1, d_2\} + 1\}$ 
4:   for all  $i \in \{1, \dots, d_3\}$  do
5:     if  $i < m$  then  $g_i^3 := g_i^2$  else  $g_i^3 := g_i^1$  end if
6:   end for
7:   for all  $i \in \{1, \dots, d_4\}$  do
8:     if  $i < m$  then  $g_i^4 := g_i^1$  else  $g_i^4 := g_i^2$  end if
9:   end for
10:  return  $\mathbf{g}^3$  and  $\mathbf{g}^4$ 
11: end procedure

```

Algorithm 8 Genetic Algorithm (GA) – Stage 1

```

1: procedure GA
2:    $P := \emptyset$ 
3:   for all  $i \in \{1, \dots, s\}$  do
4:     Randomly choose  $d_i \in \{\underline{d}, \dots, \bar{d}\}$ 
5:     Randomly choose  $\mathbf{g}^i \in \{1, \dots, n\}^{d_i}$ 
6:     if  $i = 1 \vee f(\mathbf{g}^i) < f(\mathbf{b})$  then  $\mathbf{b} := \mathbf{g}^i$  end if
7:      $P := P \cup \{\mathbf{g}^i\}$ 
8:   end for
9:   while Number of generations  $< n_g$  do
10:     $M := \emptyset$ 
11:    while  $|M| < s$  do
12:      Randomly select individuals  $\mathbf{g}^1, \mathbf{g}^2 \in P$ 
13:      if  $f(\mathbf{g}^1) \leq f(\mathbf{g}^2)$  then
14:         $M := M \cup \{\mathbf{g}^1\}$ 
15:      else
16:         $M := M \cup \{\mathbf{g}^2\}$ 
17:      end if
18:    end while
19:     $P' := \emptyset, i := 1$ 
20:    while  $i \leq s$  do
21:      if  $\text{random} < p_c$  then
22:         $(\mathbf{g}^i, \mathbf{g}^{i+1}) := \text{CROSSOVER}(\mathbf{g}^i \in M, \mathbf{g}^{i+1} \in M)$ 
23:      end if
24:       $P' := P' \cup \{\mathbf{g}^i\} \cup \{\mathbf{g}^{i+1}\}; i := i + 2$ 
25:    end while
26:    for all  $\mathbf{g}^i \in P'$  do
27:       $\mathbf{g}^i := \text{MUTATE}(\mathbf{g}^i)$ 
28:      if  $f(\mathbf{g}^i) < f(\mathbf{b})$  then  $\mathbf{b} := \mathbf{g}^i$  end if
29:    end for
30:     $P := P'$ 
31:  end while
32:  return  $\mathbf{b}$ 
33: end procedure

```

References

- [1] Adcock, S. A., 2017. GAUL: The genetic algorithm utility library.
URL http://gaul.sourceforge.net/gaul_reference_guide.html
- 25 [2] Andriosopoulos, K., Nomikos, N., 2014. Performance replication of the Spot Energy Index with optimal equity portfolio selection: Evidence from the UK, US and Brazilian markets. *European Journal of Operational Research* 234 (2), 571–582.
- [3] Beasley, J. E., 1990. OR-Library: Distributing test problems by electronic mail. *Journal of the Operational Research Society* 41 (11), 1069–1072.
- 30 [4] Beasley, J. E., Meade, N., Chang, T.-J., 2003. An evolutionary heuristic for the index tracking problem. *European Journal of Operational Research* 148 (3), 621–643.
- [5] Benidis, K., Feng, Y., Palomar, D., forthcoming. Sparse portfolios for high-dimensional financial index tracking. *IEEE Transactions on Signal Processing*.
- [6] Bertolazzi, P., Felici, G., Festa, P., Fiscon, G., Weitschek, E., 2016. Integer programming models for feature selection: New extensions and a randomized solution algorithm. *European Journal of Operational*
5 *Research* 250 (2), 389–399.
- [7] Bertsimas, D., King, A., 2015. OR forum—an algorithmic approach to linear regression. *Operations Research* 64 (1), 2–16.
- [8] Bertsimas, D., King, A., Mazumder, R., 2016. Best subset selection via a modern optimization lens. *The Annals of Statistics* 44 (2), 813–852.
- 10 [9] Brill, F. Z., Brown, D. E., Martin, W. N., 1992. Fast generic selection of features for neural network classifiers. *IEEE Transactions on Neural Networks* 3 (2), 324–328.
- [10] Busse, J. A., Goyal, A., Wahal, S., 2010. Performance and persistence in institutional investment management. *The Journal of Finance* 65 (2), 765–790.
- [11] Canakgoz, N. A., Beasley, J. E., 2009. Mixed-integer programming approaches for index tracking and
15 enhanced indexation. *European Journal of Operational Research* 196 (1), 384–399.
- [12] Chen, C., Kwon, R. H., 2012. Robust portfolio selection for index tracking. *Computers & Operations Research* 39 (4), 829–837.
- [13] Chiam, S. C., Tan, K. C., Al Mamun, A., 2008. Evolutionary multi-objective portfolio optimization in practical context. *International Journal of Automation and Computing* 5 (1), 67–80.
- 20 [14] Chiam, S. C., Tan, K. C., Al Mamun, A., 2013. Dynamic index tracking via multi-objective evolutionary algorithm. *Applied Soft Computing* 13 (7), 3392–3408.

- [15] Corielli, F., Marcellino, M., 2006. Factor based index tracking. *Journal of Banking & Finance* 30 (8), 2215–2233.
- [16] Diosan, L., 2005. A multi-objective evolutionary approach to the portfolio optimization problem. In: International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'06). Vol. 2. IEEE, pp. 183–187.
- [17] European Fund and Asset Management Association (EFAMA), 2017. 2017 EFAMA Annual Report. European Fund and Asset Management Association (EFAMA).
- [18] Filippi, C., Guastaroba, G., Speranza, M., 2016. A heuristic framework for the bi-objective enhanced index tracking problem. *Omega* 65, 122–137.
- [19] Fischetti, M., Lodi, A., 2003. Local branching. *Mathematical Programming* 98 (1–3), 23–47.
- [20] Gaivoronski, A. A., Krylov, S., Van der Wijk, N., 2005. Optimal portfolio selection and dynamic benchmark tracking. *European Journal of Operational Research* 163 (1), 115–131.
- [21] Garcia, S., Herrera, F., 2008. An extension on “statistical comparisons of classifiers over multiple data sets” for all pairwise comparisons. *Journal of Machine Learning Research* 9 (Dec), 2677–2694.
- [22] Gilli, M., Schumann, E., 2012. Heuristic optimisation in financial modelling. *Annals of Operations Research* 193 (1), 129–158.
- [23] Goldberg, D. E., 2006. Genetic algorithms. Pearson Education India.
- [24] Gottlieb, J., Julstrom, B. A., Raidl, G. R., Rothlauf, F., 2001. Prüfer numbers: A poor representation of spanning trees for evolutionary search. In: Spector, L., et al. (Eds.), *Proceedings of the 3rd Annual Conference on Genetic and Evolutionary Computation. GECCO'01*. Morgan Kaufmann Publishers Inc., San Francisco, pp. 343–350.
- [25] Guastaroba, G., Speranza, M. G., 2012. Kernel Search: An application to the index tracking problem. *European Journal of Operational Research* 217 (1), 54–68.
- [26] Investment Company Institute, 2017. 2017 Fact Book. Investment Company Institute.
- [27] Kolm, P. N., Tütüncü, R., Fabozzi, F. J., 2014. 60 years of portfolio optimization: Practical challenges and current trends. *European Journal of Operational Research* 234 (2), 356–371.
- [28] Konno, H., Wijayanayake, A., 2001. Minimal cost index tracking under nonlinear transaction costs and minimal transaction unit constraints. *International Journal of Theoretical and Applied Finance* 4 (6), 939–957.

- [29] Krink, T., Mittnik, S., Paterlini, S., 2009. Differential evolution and combinatorial search for constrained index-tracking. *Annals of Operations Research* 172 (1), 153–176.
- [30] Kuncheva, L. I., Jain, L. C., 1999. Nearest neighbor classifier: simultaneous editing and feature selection. *Pattern Recognition Letters* 20 (11), 1149–1156.
- [31] Kwiatkowski, J. W., 1992. Algorithms for index tracking. *IMA Journal of Management Mathematics* 4 (3), 279–299.
- [32] López-Ibáñez, M., Dubois-Lacoste, J., Stützle, T., Birattari, M., 2011. The irace package, iterated race for automatic algorithm configuration. Tech. rep., TR/IRIDIA/2011-004, IRIDIA, Université Libre de Bruxelles, Belgium.
- [33] Malkiel, B. G., 1995. Returns from investing in equity mutual funds 1971 to 1991. *The Journal of Finance* 50 (2), 549–572.
- [34] Maringer, D., Oyewumi, O., 2007. Index tracking with constrained portfolios. *Intelligent Systems in Accounting, Finance and Management* 15 (1–2), 57–71.
- [35] Montfort, K. v., Visser, E., van Draat, L. F., 2008. Index tracking by means of optimized sampling. *The Journal of Portfolio Management* 34 (2), 143–152.
- [36] Moral-Escudero, R., Ruiz-Torrubiano, R., Suárez, A., 2006. Selection of optimal investment portfolios with cardinality constraints. In: 2006 IEEE International Conference on Evolutionary Computation. IEEE, pp. 2382–2388.
- [37] Mutunge, P., Haugland, D., 2018. Minimizing the tracking error of cardinality constrained portfolios. *Computers & Operations Research* 90, 33–41.
- [38] Oh, I.-S., Lee, J.-S., Moon, B.-R., 2004. Hybrid genetic algorithms for feature selection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 26 (11), 1424–1437.
- [39] Raymer, M. L., Punch, W. F., Goodman, E. D., Kuhn, L. A., Jain, A. K., 2000. Dimensionality reduction using genetic algorithms. *IEEE Transactions on Evolutionary Computation* 4 (2), 164–171.
- [40] Roll, R., 1992. A mean/variance analysis of tracking error. *The Journal of Portfolio Management* 18 (4), 13–22.
- [41] Rothlauf, F., 2011. Design of modern heuristics: principles and application. Springer, Berlin, Heidelberg.
- [42] Rudd, A., 1980. Optimal selection of passive portfolios. *Financial Management*, 57–66.
- [43] Rudolf, M., Wolter, H.-J., Zimmermann, H., 1999. A linear model for tracking error minimization. *Journal of Banking & Finance* 23 (1), 85–103.

- [44] Ruiz-Torrubiano, R., Suárez, A., 2007. Use of heuristic rules in evolutionary methods for the selection of optimal investment portfolios. In: 2007 IEEE Congress on Evolutionary Computation. IEEE, pp. 212–219.
- 25 [45] Ruiz-Torrubiano, R., Suárez, A., 2009. A hybrid optimization approach to index tracking. *Annals of Operations Research* 166 (1), 57–71.
- [46] Ruiz-Torrubiano, R., Suárez, A., 2010. Hybrid approaches and dimensionality reduction for portfolio selection with cardinality constraints. *IEEE Computational Intelligence Magazine* 5 (2), 92–107.
- [47] Sant’Anna, L. R., Filomena, T. P., Caldeira, J. F., 2017. Index tracking and enhanced indexing using
30 cointegration and correlation with endogenous portfolio selection. *The Quarterly Review of Economics and Finance* 65, 146–157.
- [48] Sant’Anna, L. R., Filomena, T. P., Guedes, P. C., Borenstein, D., 2017. Index tracking with controlled number of assets using a hybrid heuristic combining genetic algorithm and non-linear programming. *Annals of Operations Research* 258 (2), 849–867.
- [49] Scozzari, A., Tardella, F., Paterlini, S., Krink, T., 2013. Exact and heuristic approaches for the index tracking problem with UCITS constraints. *Annals of Operations Research* 205 (1), 235–250.
- [50] Sharma, A., Agrawal, S., Mehra, A., 2017. Enhanced indexing for risk averse investors using relaxed second order stochastic dominance. *Optimization and Engineering* 18, 407–442.
- 700 [51] Siedlecki, W., Sklansky, J., 1989. A note on genetic algorithms for large-scale feature selection. *Pattern Recognition Letters* 10 (5), 335–347.
- [52] Skolpadungket, P., Dahal, K., Harnpornchai, N., 2007. Portfolio optimization using multi-objective genetic algorithms. In: 2007 IEEE Congress on Evolutionary Computation. IEEE, pp. 516–523.
- [53] Streichert, F., Ulmer, H., Zell, A., 2004. Comparing discrete and continuous genotypes on the constrained
705 portfolio selection problem. In: Deb, K. (Ed.), *Genetic and Evolutionary Computation GECCO 2004*. GECCO 2004. Lecture Notes in Computer Science, vol 3103. Springer, Berlin, Heidelberg, pp. 1239–1250.
- [54] Strub, O., Baumann, P., 2018. Optimal construction and rebalancing of index-tracking portfolios. *European Journal of Operational Research* 264 (1), 370–387.
- [55] Strub, O., Trautmann, N., 2016. An application of Microsoft Excel’s evolutionary solver based on a
710 novel chromosome encoding scheme to the 1/N portfolio tracking problem. In: Suryadi, K., et al. (Eds.), *Industrial Engineering and Engineering Management (IEEM)*, 2016 IEEE International Conference on. IEEE, Bali, pp. 745–749.
- [56] Strub, O., Trautmann, N., 2017. A genetic algorithm for the UCITS-constrained index-tracking problem. In: *Evolutionary Computation (CEC)*, 2017 IEEE Congress on. IEEE, pp. 822–829.

- 715 [57] Takeda, A., Niranjana, M., Gotoh, J.-y., Kawahara, Y., 2013. Simultaneous pursuit of out-of-sample performance and sparsity in index tracking portfolios. *Computational Management Science* 10 (1), 21–49.

ACCEPTED MANUSCRIPT